5-2014

A Study of Advanced Modern Control Techniques Applied to a Twin Rotor MIMO System

Andrew E. Phillips

R•I•T

A Study of Advanced Modern Control Techniques Applied to a Twin Rotor MIMO System

By

Andrew E. Phillips

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of
Master of Science in Electrical Engineering

Kate Gleason College of Engineering
Department of Electrical and Microelectronic Engineering

DEPARTMENT OF ELECTRICAL AND MICROELECTRONIC ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK 14623
MAY 2014

A Study of Advanced Modern Control Techniques Applied to a Twin Rotor MIMO System

By

Andrew E. Phillips

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Electrical Engineering

Approved By:

Dr. Ferat Sahin – *Thesis Advisor* _____

Department of Electrical and Microelectronic Engineering

Dr. Vincent Amuso – *Associate Professor* _____

Department of Electrical and Microelectronic Engineering

Dr. Agamemnon Crassidis – *Associate Professor* _____

Department of Mechanical Engineering

Dr. Sohail A. Dianat – *Department Head* _____

Department of Electrical and Microelectronic Engineering

DEPARTMENT OF ELECTRICAL AND MICROELECTRONIC ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK 14623

February 2014

Abstract

The twin rotor MIMO system (TRMS) is a helicopter-like system that is restricted to two degrees of freedom, pitch and yaw. It is a complicated nonlinear, coupled, MIMO system used for the verification of control methods and observers. There have been many methods successfully applied to the system ranging from simple proportional integral derivative (PID) controllers, to machine learning algorithms, nonlinear control methods and other less explored methods like deadbeat control and various optimal methodologies. This thesis details the design procedure for two different control methods. The first is a suboptimal tracking controller using a linear quadratic regulator (LQR) with integral action. The second is the design of several adaptive sliding mode controller to provide robust tracking control of the TRMS. Once the design is complete the controllers are tested in simulation and their performance is compared against a PID controller experimentally. The performance of the controllers are also compared against other controllers in the literature. The ability of the sliding mode controllers (SMC) to suppress chattering is also be explored.

Acknowledgements

I would like to thank Dr. Ferat Sahin for his support in pursuing this research. I would also like to thank Dr. Agamemnon Crassidis for his help in implementing the sliding mode controller experimentally. I would like to thank James Stephano for his assistance in setting up the required hardware. And finally I would like to thank all of my professors for all their help over the years.

Table of Contents

List of Figures

List of Tables

1 Background

This chapter provides a survey of previous work done in relation to the twin rotor MIMO system. As well as a description of the work that is done in this thesis.

1.A. Introduction

The TRMS is a common control problem for validating new control methodologies. The simplest solution to the problem is to apply PID control to the system. There has also been a lot of research into applying machine learning to the system to find the optimal parameters to guarantee the best performance. However these machine learning algorithms do not ensure robust performance, which is desired in aerospace applications. To ensure robust performance various control methods such as deadbeat control, H_∞ and SMC has been applied to the system as well, albeit to a lesser extent.

The simplest solution to most any control problem is to use a hand tuned PID controller, the TRMS is no exception. However in an effort to take the human out of the slow tuning process the PID controller is typically tuned via a machine learning process. In 2012 Meon M.S. et al proposed a method called PID Active Force Control [1] in which a system estimated the external torque disturbances and a neural network and fuzzy logic were used to optimize the PID controller. The method provided a smooth, albeit slow, response that worked well at rejecting disturbances. However the controller was only implemented in simulation and the results for the yaw subsystem were never reported [1].

A more complex option is to use a Linear Gaussian (LG) based controller which has the benefit of having its own robustness properties [2] [3] [4]. There are two forms of the LG controllers that have been implemented in simulation. The first is the LQR design methodology, which requires a linear system, the TRMS is a nonlinear system so the system must be put into a linear form. It also requires full state feedback so a suitably robust observer must be implemented. The second form is the *Linear Quadratic Gaussian* (LQG) controller which combines the linear quadratic regulator (LQR) controller with a Kalman filter. The advantage of this method is that an optimal estimator is used to provide the full state feedback. Again the system must be put into a linear form to utilize this method.

In 2004 A.Q. Khan used the LQR method to control a 3 DOF helicopter [5]. The steady state solution was applied to the linearized system model to implement full state feedback. The controller provided adequate performance around the equilibrium point. However the controller was not considered robust in the face of uncertainties and future work was needed to make it so. In 2012 B. Pratap demonstrated a new approach to the LQR problem applied to the TRMS in simulation. The system was decoupled into two subsystems and then linearized [6] an approach that was made use of in other implementations [7] [8] [9] [10]. Two LQR controllers were designed for the subsystems, however instead of taking the suboptimal solution to the LQR problem the Kalman gain was updated iteratively until the cost function was minimized. The controller was used simply as a regulating controller though, so no conclusion could be drawn on the tracking results [6].

Another approach to the optimal control problem is to use LQG controllers. In 2000 S.M. Ahmed et al applied the LQG method to a one degree of freedom (DOF) helicopter [11]. The LQG compensator failed to provide adequate performance alone so a prefilter was used to attenuate high frequency vibration and lower the command effort required. In 2013 a thesis by A.K. Agrawal showed through simulation that a LQG controller with weighting matrices optimized using a bacterial foraging method to control the TRMS [12]. The resulting controller showed improved performance over the controller designed using manually weighted matrices.

Sliding mode control is another form of robust control with the added advantage of being applied to nonlinear systems. It also offers the advantage of being able to guarantee robustness over a known operating range and in the face of known bounded disturbances. However in order to ensure perfect tracking using the generic SMC design methodology the system must be square, meaning it must have the same number of inputs as states. The TRMS has only two inputs and six or seven states (depending on the model) making it underactuated. The general solution to this problem is to decouple the system into two single input single output (SISO) systems [13] [14] [9] [10] [8].

Following the decoupling method, in 2011 L. Huang decoupled the system into the two subsystems and simplified the system by making several modeling assumptions and linearizing the system to create two SISO SMC controllers where the switching gain was calculated based on the assumptions made in the controller design [15]. However it appeared that the resulting

system performance in hardware was slow and prone to small oscillations about the desired set point. Also the switching gain did not take into account parameter variation or disturbances, thus robustness cannot be guaranteed over different operating conditions. It was believed that improving the state feedback would improve performance in future systems [15].

One of the challenges of implementing SMC in hardware is the chattering that is inherent with the discontinuity of the control law. There are many methods to overcome this problem. One method used to reduce chattering that is characteristic of SMC is to use a higher order sliding surface. This has the effect of integrating the discontinuity from the control law thus making the input continuous. In 2010, Q. Ahmed et al implemented a second order SMC in simulation to control the TRMS in order to eliminate the chattering on the input [7]. The controller was implemented in hardware and was able to control the TRMS but again the system was slow. The inputs showed little improvement to chatter and no results were shown for two degrees of freedom. In 2011, S. Mondal et al created a second order SMC to control the TRMS. Again the system was decoupled into a vertical and horizontal system then linearized, treating the unmodeled dynamics as disturbances [9]. The system eliminated the chatter in simulation but was never implemented in hardware. Again in 2011 S. Mondal et al continued work on the system to make the controller adaptive to account for the unmodeled disturbances and dynamics. However again the controller was never implemented in hardware. The adaptive tuning law implemented in this thesis is based off of the work done by S. Modal et al.

A second solution to attenuating the chattering of SMC is to use a boundary layer. In 2013, D.K. Saroj et al implemented a SMC with a Luenberger type non-linear observer to allow for better tracking of the TRMS [14]. The controller allowed for excellent tracking of the both the system states and the reference signal, the system was again, slow, with a settling time of about 20 seconds for the pitch and 15 seconds for the yaw. There was nothing published on the controllers ability to attenuate the input chattering.

More recently there has been attempts to integrate SMC with machine learning algorithms such as fuzzy logic and particle swarm optimization. In 2012 F. Allouani et al used particle swarm optimization (PSO) to tune the fuzzy logic membership functions and the parameters of the SMC sliding surface to achieve optimal results [13]. The simulation results were excellent however they were never implemented in hardware so it is unknown if the controller is practical.

The system input was also never shown so it is unknown if the method was able to reduce system chatter at all.

1.B. Current Work

In this thesis a suboptimal controller solution as well as a sliding mode controller is proposed, designed, simulated and implemented in hardware. The TRMS system of equations are derived and then linearized about the equilibrium point. A LQR controller is designed, the design process is modified to impart integral action onto the loop.

Three SMC are created and compared. The first uses a boundary layer, the second uses a quasi-switching function and the third is a second order SMC. The ability of these controllers to suppress chattering will be compared. All three controllers make use of an adaptive control law proposed by S. Mondal et al. The TRMS model is simplified and then decoupled into a vertical and horizontal subsystem, the two subsystems are used in the controller design. To provide full state feedback a Luenberger observer is used. The controller is simulated in MATLAB and then implemented in hardware. The performance of the controller is compared against that which is achieved using PID controllers for both a step input and a tracking input, as well as other controllers implemented in the literature.

In chapter two the theoretical background of the PID, LQR and SMC is explained. In chapter three the TRMS model is derived and its stability is examined. In chapter four the LQR and sliding mode controllers are derived. In chapter five the simulation results are examined and are compared against those achieved in the literature. In chapter six the experimental results are examined and compared against those in the literature. In chapter seven the results of the controllers are considered and future work is proposed.

2 Theoretical Background

In this chapter the theoretical background for the PID, LQR and sliding mode controllers are derived and explored. First the PID controller is explained. Next the LQR controller is explained. First the equations are derived, then suboptimal control is introduced, next it is explained how integral action can be imparted on the controller, finally its robustness properties are explored. The finally the concept of SMC is explained using an example.

2.A. PID Control

The simplest way of implementing a controller is by hand tuning a PID controller. It requires no knowledge of the system and it can stabilize most any system that is controllable. There are three principal ways of looking at PID controller. The first way is the classical control methodology. The PID controller adds two poles and two zeroes to a system that can be used to cancel existing poles or to insert new dominant poles into the system [16]. The second way is related to the first way, it weights the different frequency components of the error based on the location of the PID controller's poles and zeroes. The final way is that it creates a new input based on the weighted sum of the error, the integral of the error, and the rate of change of the error.

While PID controllers are a simple method of controlling a system, there are many better methods. These include but are not limited to modern control, deadbeat control, robust control, nonlinear control, and optimal control. These methods are generally favored (among other reasons) because it is widely recognized that full state feedback provides better performance than output feedback. Optimal control is used for several reasons including, but not limited to, robustness, and low control effort. The next section explains the principals of the LQR.

2.B. LQR Control

Designing a controller for a system by placing the poles results in a system of equations that is *overdetermined*; that is there are more equations than variables to be solved for. Because there is more than one solution to this problem there must be one that is quantifiably better than the others. This gave birth to the concept of Optimal Control Theory. There are many solutions to the optimal control theory such as LQR, LQG, and dynamic programming.

2.B.i. Optimal Control

All optimal control problems make use of a cost function, a function that is positive for all values and ever increasing. It is defined from variables inherent to the system (the input and output) as well as designer defined values that determine what is more important about the way the system responds. Considering the continuous linear system shown in Equation (1), a cost function can be defined as Equation (2).

$$\dot{x} = Ax + Bu \tag{1}$$

$$J = \frac{1}{2}x_f^T S_f x_f + \frac{1}{2}\int_{t_0}^{t_f} x^T Q x + u^t R u dt \tag{2}$$

Here x_f is the final state of the system, x are the states of the system over time, u is the input to the system over time, S is a positive definite time varying matrix that weights the final state of the system, Q is a positive definite matrix that weights the states of the system over time and R is positive semi-definite matrix that weights the input of the system over time. The goal is to find some input that minimizes Equation (2) subject to Equation (1). Using Lagrangian operators Equation (3) is formed, this is the augmented performance function.

$$J' = \frac{1}{2}x_f^T S_f x_f + \frac{1}{2}\int_{t_0}^{t_f} x^T Q x + u^t R u + \lambda^T (Ax + Bu - \dot{x}) dt \tag{3}$$

To find a solution several assumptions are made. First Q, R, and S are symmetrical and second that $\lambda_f = S_f x_f$. From Equation (3) the Hamiltonian is defined as Equation (4).

$$H = \frac{1}{2}(x^T Q x + u^T R u) + \lambda^T (Ax + Bu - \dot{x}) \tag{4}$$

Taking the derivative of Equation (4) with respect to the optimization parameters λ and x Equations Equation (5) through Equation (7) are found.

$$\dot{x} = Ax + Bu \tag{5}$$

$$-\dot{\lambda} = A^T \lambda + Qx \tag{6}$$

$$Ru + B^T \lambda = 0 \tag{7}$$

Rewriting Equation (7) using the assumption about λ the input equation, Equation (8) is found. K is called "The Kalman Gain".

$$u = -R^{-1}B^T Sx = -Kx \tag{8}$$

Using the assumptions from before with Equations (5) and (6) the Algebraic Riccatti Equation is found to be Equation (9).

$$-\dot{S} = A^T S + SA - SBR^{-1}B^T S + Q \tag{9}$$

Equation (9) is time varying so it provides a set of differential equations that must be solved in conjunction with the required control effort. However this is not always be the case. If a steady state solution to Equation (9) is found then that can be used instead of the time varying solution. This is called the *suboptimal solution*.

2.B.ii. Suboptimal Control

It can be seen from Equations (8) and (9) that the Kalman Gain is time varying. However the optimal Kalman Gain matrix can be approximated in what is called "the suboptimal solution". Consider the discrete system shown in Equation (10). The initial state is given as $x_0 = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$.

$$\dot{x} = \begin{bmatrix} .5183 & .1173 \\ -.5867 & .5183 \end{bmatrix} x + \begin{bmatrix} 5.1070 \\ -1.9048 \end{bmatrix} u \tag{10}$$

The discrete equivalent of the optimal control Equations (11) and (12).

$$u_k = -(R - B^T S_{k+1} B)^{-1} B^T S_{k+1} A x_k = -K_k x_k \tag{11}$$

$$S_k = A^T S_{k+1}[I - B(B^T S_{k+1} B + R)^{-1} B^T S_{k+1}] + Q \tag{12}$$

Using Equations (11) and (12) a digital LQR controller was found and the Kalman gain values over time were plotted using MATLAB. The results are shown in Figure 1. The MATLAB code used to create this plot is provided in Appendix A.

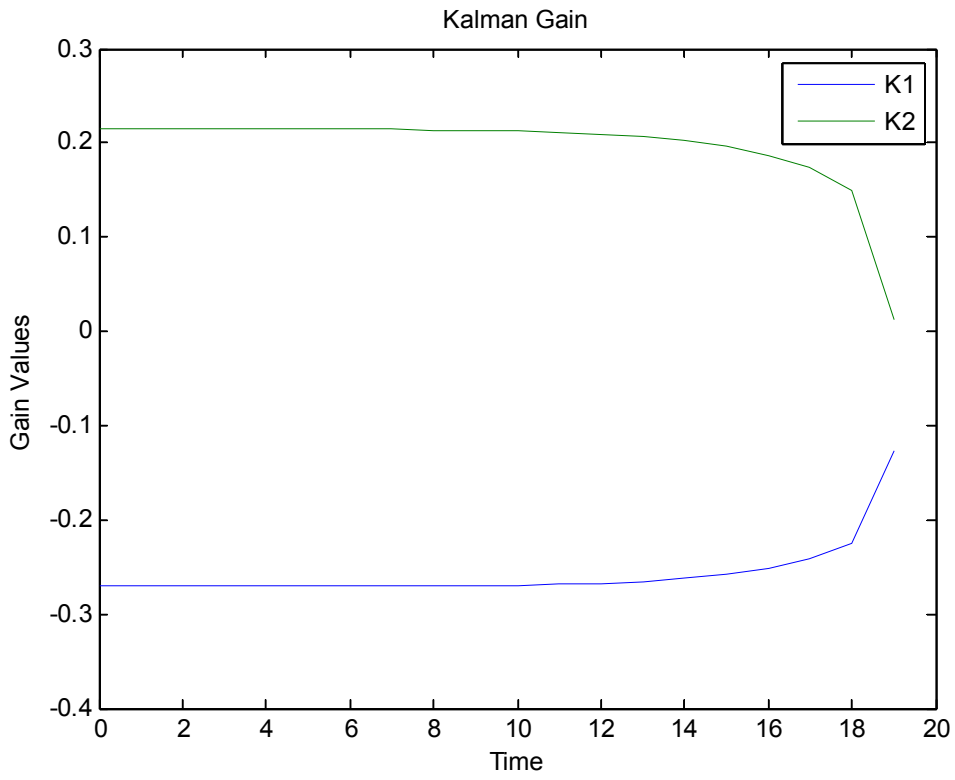

Figure 1: Example of the Kalman gain changing over time.

It can be see that for the majority of the time the Kalman gain is at steady state. Because of this a new gain matrix can be defined called the suboptimal feedback matrix. It is defined in Equation (13).

$$K_{ss} = K(t_0) \tag{13}$$

The downside to the LQR method is that integral action is not imparted on the loop by default. However there is a method that can be used to guarantee that integral action is imparted on the loop. This is described in the next section.

2.B.iii. Adding Integral Action

In many situations it is desired that the control loop has integral action to guarantee no steady state error to a step input. To impart integral action on the loop, the system is restated in a form that creates a number of additional states equal to the number of outputs that are the output error of the system. Figure 2 shows the block diagram of the LQR controller with integral action (LQI).

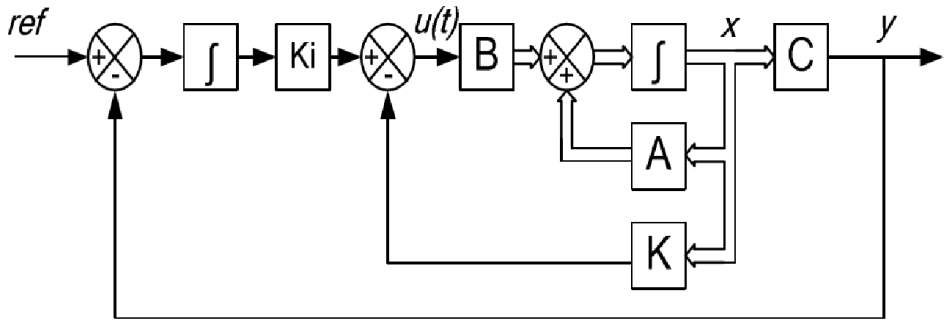

Figure 2: Block diagram of suboptimal LQR control with integral action [17].

By augmenting the state space system as shown in Equations (14) and (15), integral action will be imparted onto the loop [18].

$$\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \tag{14}$$

$$\hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \tag{15}$$

The A matrix must remain square, so from inspection it can be seen that the effect of this augmentation is the addition of a number of poles at the origin equal to the number of outputs. The suboptimal solution to the augmented system can be solved the same way as detailed in part 2.B.ii.

In addition to integral action another characteristic that is desired of a controller is robustness. Robustness allows a system to continue to function properly in the face of changes of system parameters or dynamics. There are four types of control theory that can guarantee robustness: deadbeat, robust control theory, sliding mode control and LG based controllers.

2.B.iv. Robustness Properties

It is well known that the LQ controllers exhibit robust properties [4] [3]. This can be shown by considering Equation (16), the input to the system is perturbed by some complex value α [2].

$$u = -\alpha K x \tag{16}$$

To show the system remains stable for variation in the value α, a Lyapunov candidate is defined in Equation (17).

$$V = \frac{1}{2} x^T S x \tag{17}$$

Taking the derivative of Equation (17) with respect to time, and assuming that a steady state solution is used for the Kalman gain.

$$\dot{V} = \dot{x}^T S + S\dot{x} \tag{18}$$

Substituting Equation (1) and Equation (16) into Equation (18) yields Equation (19).

$$\begin{aligned} \dot{V} &= [(A + \alpha BK)x]^T S + S(A + \alpha BK)x \\ &= x^T (A^T S + SA - \alpha K^T B^T S - \alpha SBK)x \end{aligned} \tag{19}$$

Assuming a steady state solution to Equation (9) and substituting it into Equation (19) yields Equation (20).

$$\dot{V} = x^T [K^T R K - Q - \alpha (K^T B^T S - SBK)]x \tag{20}$$

It is known that $K = R^{-1} B^T S$, and $K^T R K = S B R^{-1} B^T S$ [19], substituting these into Equation (20) yields Equation (21).

$$\dot{V} = -x^T (Q + (2\alpha - 1) K^T R K)x \tag{21}$$

It is known that Q ≥ 0 and $K^T R K \geq 0$, so if α > ½ then stability can be guaranteed. It is simple to see that if the magnitude of α is greater than ½ then the system is stable. The proof of the stability of the system with a phase disturbance is harder to show. If α is complex then it can be expressed as Equation (22).

$$Re\{e^{j\Phi}\} = \cos(\Phi) \tag{22}$$

So as long as |Φ| ≤ 60° then $\cos(\Phi) > \frac{1}{2}$ and the system will remain stable. This means that ideally LQ controllers can tolerate gain attenuation of between ½ and ∞ and a phase disturbance between ±60°. However as [4] showed it is possible to choose Q and R values such that these conditions are not met [4].

As mentioned before another method of robust control is sliding mode control. An advantage of sliding mode control is that it can be applied directly to nonlinear systems. This is the subject of the next section.

2.C. Sliding Mode Control

Sliding mode control is designed differently from most controllers. Instead of pole placement a Lyapunov approach is used to show that a controller guarantees stability over an operating range. In order to derive sliding mode control the concept of nonlinear stability must be explained first. This is done in the form of the Lyapunov stability theorem.

2.C.i. Lyapunov Theory

In linear systems the stability of a system is determined by the real part of its poles (or eigenvalues). Because of this it is a simple task to determine a system's stability, both in open and closed loop configurations. Determining the stability of nonlinear systems is a considerably more difficult task.

One way is using a method called phase portraits. The idea is to create a plot where the states are the axis of the graph. It allows for one to see characteristics of the system such as equilibrium points, limit cycles, and saddle points. Phase portraits are produced by setting the input to a system to zero and then applying some set of initial conditions and allowing the system to produce its natural response [20]. Figure 3 shows three examples of second order phase portraits.

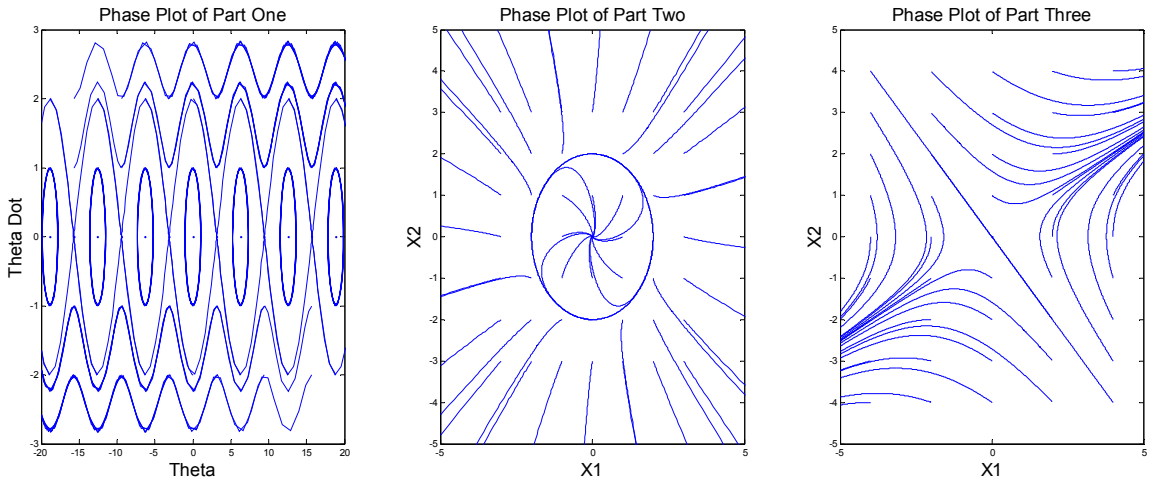

Figure 3: Three examples of phase portraits, (3a) left, (3b) center, and (3c) right.

Figure 3a was produced by Equation (23). It can be seen that it contains seven equilibrium points placed at $\theta = n\pi$ where n is some integer.

$$\ddot{\theta} + \sin(\theta) = 0 \tag{23}$$

Figure 3b shows a system produced by the system of equations shown in Equation (24). It has an equilibrium point at zero and a possible limit cycle. The area within the limit cycle is stable and the area outside the limit cycle is unstable.

$$\begin{aligned} \dot{x}_1 &= x_1(x_1^2 + x_2^2 - 4) + x_2 \\ \dot{x}_2 &= x_2(x_1^2 + x_2^2 - 4) - x_1 \end{aligned} \tag{24}$$

Figure 3c was produced by the system of equations in Equation (25). It contains a saddle point. It can be inferred that if the initial conditions of the system lie along the diagonal line then the initial conditions cancel the unstable poles of the system resulting in a stable trajectory.

$$\ddot{x} + \dot{x} - 2x = 0 \tag{25}$$

The downside to this method is that if a set of initial conditions are not fed into the phase portrait nothing can be said about the response to that set of initial conditions. In addition, it does not say anything about the input to the system, or allows for the system to be analyzed mathematically. Lyapunov stability theorem allows for one to prove that a system is stable whether it is linear or nonlinear [21].

The idea behind Lyapunov stability theorem is that if the energy of a system is constantly decreasing then the system will return to its equilibrium point. If $\dot{\vec{x}} = f(\vec{x})$ where $f(\vec{x})$ belongs to the domain *D*, and a function V belongs to the same domain *D* such that

1. $V(\vec{x} = 0) = 0$
2. $V(\vec{x}) > 0 \quad \forall \vec{x} \neq 0$
3. $\dot{V}(\vec{x}) < 0$
4. $V(\vec{x})$ is radially unbounded

If 1 through 4 are true then $f(\vec{x})$ is globally asymptotically stable. However if any of these are not true then it does not prove that the system is unstable. The sliding mode controller is designed using this principal. An example is provided in the next session.

2.C.ii. Sliding Mode Control

Sliding mode control is based off the idea of variable structure control. The simplest way to explain is to provide an example. Consider the plant equation shown in Equation (26).

$$bu(t) = m\ddot{x} + c\dot{x} + kx \tag{26}$$

Equation (26) is a simple mass spring damper system where *c* is the system damping, *m* is the mass, *k* is the spring constant, and *b* is some input scaling constant. The state *x* is the position of the mass.

The first step is to define something called a "sliding surface". The sliding surface is a set of dynamics defined by the designed that the controller will force the system to act as [22]. Equation (27) shows a general form for the sliding surface where *n* is the order of the sliding surface (not the system order), λ is some positive constant and $\tilde{x} = x - x_d$ where x_d is the desired state. In this application a first order controller will be used.

$$S = (\lambda + \frac{d}{dt})^n \tilde{x} \tag{27}$$

A sliding mode controller has two phases, the first is called the reaching phase. During this phase the controller is guiding the system state onto the sliding surface. The second phase is called the sliding phase, during which the system state 'slides' along the sliding surface. Figure 4 shows the phase plane of a second order sliding surface.

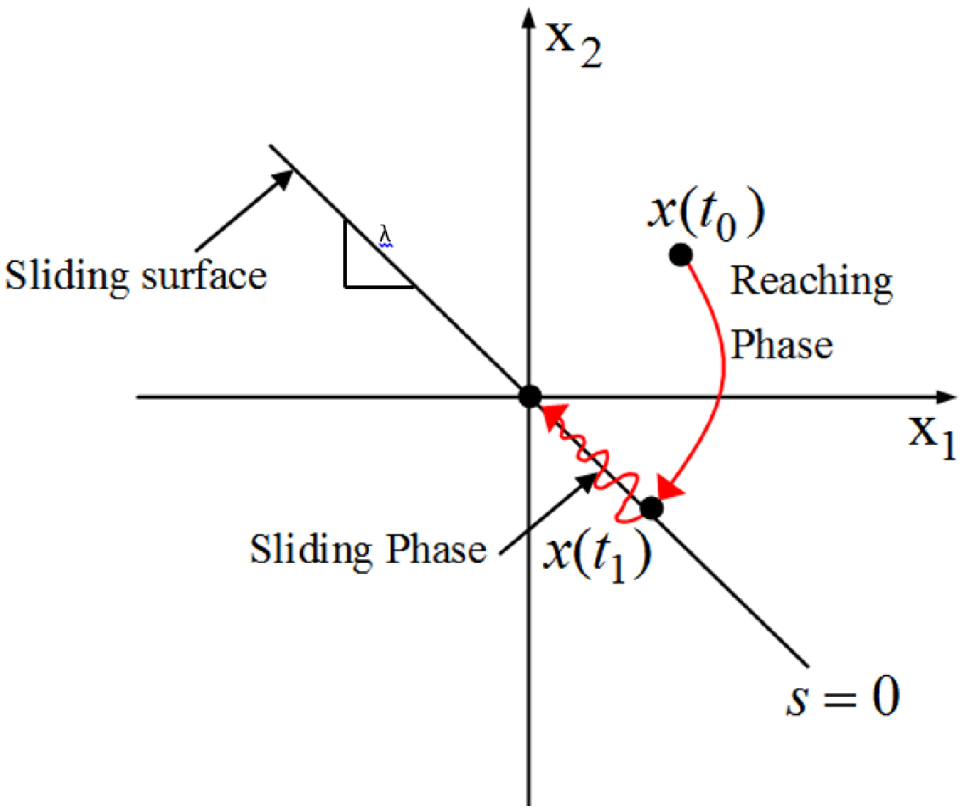

Figure 4: Generic first order sliding surface [23].

To enforce the sliding phase, the derivative of Equation (27) is taken with respect to time (remember it is a first order controller) and setting it equal to zero. It is important that the highest order derivative of the plant appears in Equation (28).

$$\dot{S} = \ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}} = 0 \tag{28}$$

Substituting Equation (26) into Equation (28) yields Equation (29).

$$\dot{S} = \frac{1}{m}[bu(t) - c\dot{x} - kx] - \ddot{x}_d + \lambda\dot{\tilde{x}} = 0 \tag{29}$$

The derivative of the sliding surface is equated to what is called the *reaching law*. The reaching law enforces the robustness of the controller.

$$\dot{S} = -\eta sign\,(S) \tag{30}$$

Solving Equation (29) with Equation (30) for the input yields Equation (30), this is the control law.

$$u(t) = \frac{1}{\hat{b}}\{\hat{m}[\ddot{x}_d - \lambda\dot{\tilde{x}} - \eta sign\,(S)] + \hat{c}\dot{x} + \hat{k}x\} \tag{30}$$

From Figure 4 it can be seen that if the plant state overshoots the sliding surface it must be forced back onto it. This is what the discontinuous term is responsible for. An ideal discontinuity is used, the *sign()* function and it is multiplied by some small positive constant η. In addition the average value of the system parameters are used, these are denoted by the '$\hat{}$'.

Equation (30) is the proposed input to the system. To prove that Equation (30) stabilizes the system, and to determine the minimum value of the switching gain the Lyapunov stability theorem is used. A Lyapunov candidate is defined for a scalar system as Equation (31).

$$V = \frac{1}{2}S^2 \tag{31}$$

Taking the derivative of Equation (31) with respect to time we get Equation (32).

$$\dot{V} = S\dot{S} \tag{32}$$

Substituting Equation (28) into Equation (32) we get Equation (33).

$$\dot{V} = S(\ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}}) \tag{33}$$

Substituting Equation (26) into Equation (33) yields Equation (34).

$$\dot{V} = S\{\frac{1}{m}[bu(t) - c\dot{x} - kx] - \ddot{x}_d + \lambda\dot{\tilde{x}}\} \tag{34}$$

Substituting Equation (30) into Equation (34) yields Equation (35).

$$\dot{V} = S\{\frac{1}{m}[b\frac{1}{\hat{b}}\{\hat{m}[\ddot{x}_d - \lambda\dot{\tilde{x}} - \eta sign\,(S)] + \hat{c}\dot{x} + \hat{k}x\} - c\dot{x} - kx] - \ddot{x}_d + \lambda\dot{\tilde{x}}\} \tag{35}$$

If the system parameters are equal to the average values then Equation (34) reduces so Equation (35).

$$\dot{V} = -S\eta sign(S) = -\eta|S| \leq 0 \tag{36}$$

So the ideal controller stabilizes the system. However what if the system parameters are not equal to the average values? Rewriting Equation (29) as Equation (37) and switching notation from η to *Ksg*.

$$\dot{S} = \left(\frac{b\hat{c}}{\hat{b}} - c\right)\dot{x} + \left(\frac{b\hat{k}}{\hat{b}} - k\right)x + \left(\frac{b\hat{m}}{\hat{b}} - m\right)(\ddot{x}_d - \lambda\dot{\tilde{x}}) - \frac{b}{\hat{b}}K_{sg}sign\,(S) \tag{37}$$

Multiplying both sides by *S* Equation (38) is formed.

$$\dot{V} = S\dot{S} \leq -\eta|S| \tag{38}$$

Substituting Equation (37) into Equation (38) we get Equation (39).

$$S[\left(\frac{b\hat{c}}{\hat{b}} - c\right)\dot{x} + \left(\frac{b\hat{k}}{\hat{b}} - k\right)x + \left(\frac{b\hat{m}}{\hat{b}} - m\right)(\ddot{x}_d - \lambda\dot{\tilde{x}}) - \frac{b}{\hat{b}}K_{sg}sign\,(S)] \leq -\eta|S| \tag{39}$$

By taking the absolute value of both sides to enforce the worst case condition and solving for the switching gain Equation (40) is formed. So for stability over parameter variation the switching gain is defined as Equation (40).

$$K_{sg} \geq \left|\hat{c} - \frac{c\hat{b}}{b}\right||\dot{x}| + \left|\hat{k} - \frac{k\hat{b}}{b}\right||\dot{x}| + \left|\hat{m} - \frac{m\hat{b}}{b}\right|\left|\ddot{x}_d - \lambda\dot{\tilde{x}}\right| + \frac{\hat{b}}{b}\eta \tag{40}$$

Where m, c, k, and b are unknown but bounded. Thus the new input is defined in Equation (41).

$$u(t) = \frac{1}{\hat{b}}\{\hat{m}[\ddot{x}_d - \lambda\dot{\tilde{x}} - K_{sg}sign\,(S)] + \hat{c}\dot{x} + \hat{k}x\} \tag{41}$$

If it is desired to add integral action to the loop Equation (27) can be modified [22]. Equation (42) shows this modification.

$$S = \left(\lambda_1 + \frac{d}{dt}\right)^n \tilde{x} + \lambda_2 \int_0^t \tilde{x}dt \tag{42}$$

A significant problem with the sliding mode controller is that it requires infinite bandwidth. The *sign()* function cannot be implemented in the real world because it requires that the control hardware reacts instantly. There are several methods that can be used to mitigate this.

2.C.iii. Chatter Suppression Techniques

Chattering is inherent in SMC due to the discontinuous term, *sign()*. When the sliding surface crosses over zero the sign will change causing the $K_{sw}sign(s)$ term to change rapidly. This will result in the sliding surface to reverse direction and cross back over zero, which in turn causes the $K_{sw}sign(s)$ term to change again. This is the phenomenon called chattering and is caused by the fact that SMC requires infinite bandwidth. Unfortunately we cannot provide infinite bandwidth, so in order for the controller to be implemented practically the chattering has to be attenuated.

There are several methods for attenuating the chattering including using a boundary layer, replacing the switching function, and higher order sliding mode controllers. The simplest method is to low pass filter the input to reduce the chattering [24]. The downside to this is that it inserts phase lag into the controller.

Another solution, called the boundary layer is implemented by replacing the *sign(s)* term with a saturation function, Equation (43). The limits of the saturation function determine the limits of the boundary layer and ϕ determines the slope of the term when it is within the boundary layer [22]. Work has been done to make the boundary layer with variable thicknesses [25], time dependent or even state dependent [26].

$$sign(s) \rightarrow sat(\frac{s}{\phi}) \tag{43}$$

Another method is called quasi-sliding mode control (QSMC). With this method the *sign()* term is replaced with a continuous term, *sigmoid()* [27]. The *sigmoid()* function has the advantage of being a continuous function and is not subject to the abrupt switching that happens with the *sign()* function. Equation (44) shows the sigmoid function. Here, *a* is some positive constant and *S* is the input.

$$y(S) = \frac{S}{|S| + a} \tag{44}$$

The last solution mentioned here is second order sliding mode control (SOSMC). In this method the second derivative of the sliding variable S is taken and used to solve for the derivative of the input. This causes the discontinuities to be integrated resulting in greatly reduced chattering. The downside to this method is that it requires a more complicated control law.

In this thesis three methods were explored in order to reduce the chattering. The first is the boundary layer, the second is the quasi sliding mode controller and the third is the SOSMC. The ability for the controllers to suppress chatter and provide adequate performance will be compared.

Both the LQR and SMC techniques require full state feedback. To complicate things the system state may not be always be directly measureable, or easily measureable. To solve this problem the plant model can be used to estimate the system state. When this is done the result is

called an observer. The simplest type of observer is called the Luenberger Observer and is the subject of the next section.

2.D. Luenberger Observer

It is well known that controllers that implement full state feedback generally provide better performance than those that only use output feedback, both SMC and the LQR are examples of controllers that require full state feedback. The issue with this is that it is very rare that it is possible to measure all the states of a system directly, it is for this reason that the concept of the observer was created. An observer is a model of a plant that is run in parallel with the actual plant in order to determine the internal system state [28]. Consider a plant that is described completely using Equation (45).

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{45}$$

A linear model of the plant is created and is shown in Equation (46), here '$\hat{}$' denotes the best guess.

$$\begin{aligned} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u \\ \hat{y} &= \hat{C}\hat{x} + \hat{D}u \end{aligned} \tag{46}$$

Equation (46) is used to design a full state feedback matrix K to control the plant. Figure 5 shows the state space system being controlled using the full state feedback matrix K.

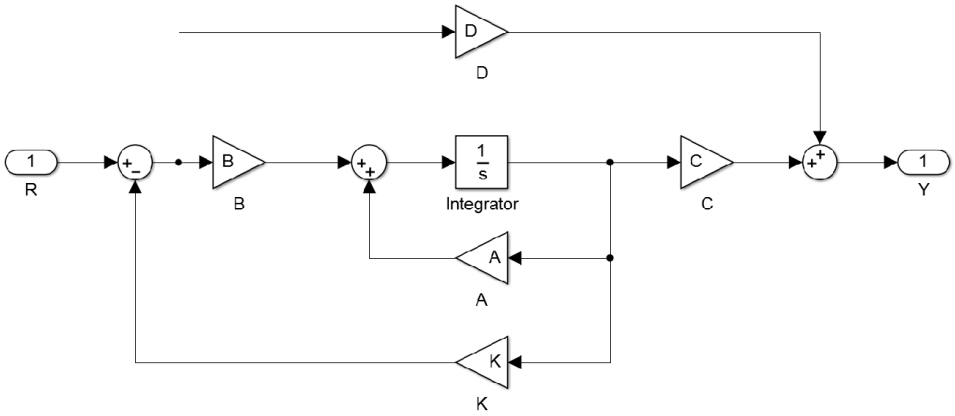

Figure 5: State space with full state feedback.

Equation (47) shows the closed loop dynamics of the state space system with full state feedback. Here 'r' is the reference signal.

$$\begin{aligned} \dot{x} &= (A - BK)x + Br \\ y &= (C - DK)x + Dr \end{aligned} \tag{47}$$

However, as stated earlier, all the states of the system may not be available for measurement. The state space model of the plant shown in Equation (46) can be used to simulate the response of the system based on the known input to the system. Figure 6 shows the state space system being controlled using full state feedback taken from the model of the plant.

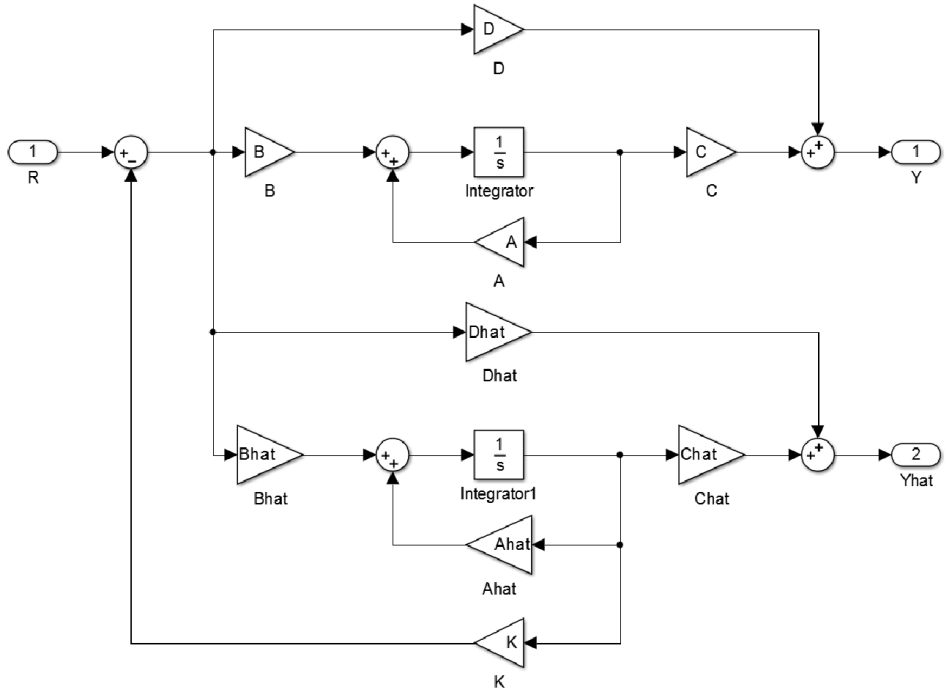

Figure 6: State space system using full state feedback via an open loop observer.

The problem with this method is that the observer is being run open loop. Because of this the observer is not robust to disturbances or parameter variations the actual plant is subject to and will lose tracking. To mitigate this, the observer can be stabilized using an output feedback matrix *L*. This increases the observer's robustness in the face of disturbances and/or parameter variation of the plant. Figure 7 shows the structure of the closed loop observer, it is also referred to as the Luenberger observer.

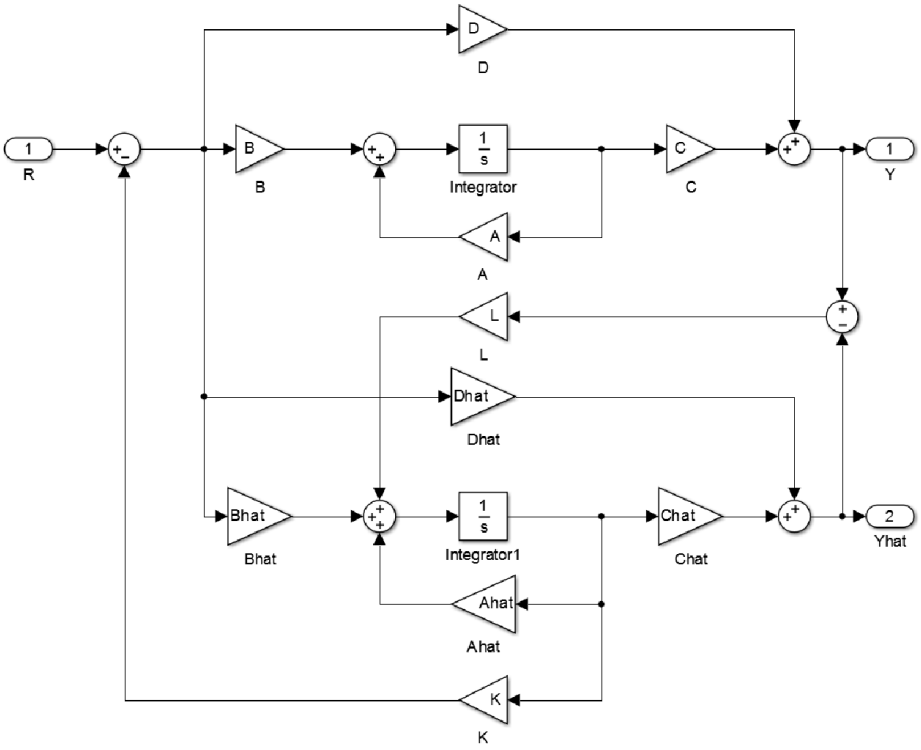

Figure 7: Closed loop state space system utilizing a Luenberger observer.

The closed loop dynamics of the Luenberger observer are shown in Equation (48). If the matrix L is chosen such that $(\hat{A} - L\hat{C})$ is stable then the error of the observer will tend to 0.

$$\begin{aligned} \dot{\hat{x}} &= (\hat{A} - L\hat{C})\hat{x} + \hat{B}u + L(y - \hat{y}) \\ \hat{y} &= \hat{C}\hat{x} + \hat{D}u \end{aligned} \tag{48}$$

Looking at Figure (8) it is obvious that the matrix L will affect the closed loop dynamics of the system as a whole. Equation (49) shows the dynamics of the closed loop system in block matrix form.

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} \hat{A} & -BK \\ L\hat{C} & A_{OC} - L\hat{D}K \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} \hat{B} \\ B - L(D - \hat{D}) \end{bmatrix} r \\ \hat{y} &= [\hat{C} - \hat{D}K] \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \hat{D}r \end{aligned} \tag{49}$$

Where A_{OC} is given by Equation (50).

$$A_{OC} = A - LC + LDK - BK \tag{50}$$

There are several things worth discussing further. First of all this method is meant for linear systems, however because of the presence of the feedback matrix it is capable of handling

non-linearity to a certain degree. The second thing is that the observer must remain stable if the output from the plant becomes disconnected from the plant. Typically if the L or K gains are too high the system will become unstable if the output is disconnected.

Now that the theoretical background has been explained the next step is to examine the TRMS. This is done in the next section.

3 The Twin Rotor MIMO System

This section contains the derivation of the plant model as well as a stability analysis if the TRMS. At the end of the section the step to setting up the TRMS are explained in detail to make it easier to replicate the work done here.

3.A. Derivation of Plant Model

The TRMS is a laboratory setup provided by Feedback Instruments for the purpose of testing new controllers. The TRMS is characterized by unstable, highly coupled, non-linear dynamics. The setup consists of a horizontal beam fixed to a vertical pillar via a two dimensional pivot. The main rotor is affixed to the front of the horizontal beam parallel to the ground. The tail rotor is affixed to the rear of the beam perpendicular to the ground. A counterbalance beam is affixed to the horizontal beam at the pivot to move the equilibrium point of the system. The main and tail rotors are controlled by two DC motors. The setup can be seen in Figure 8.

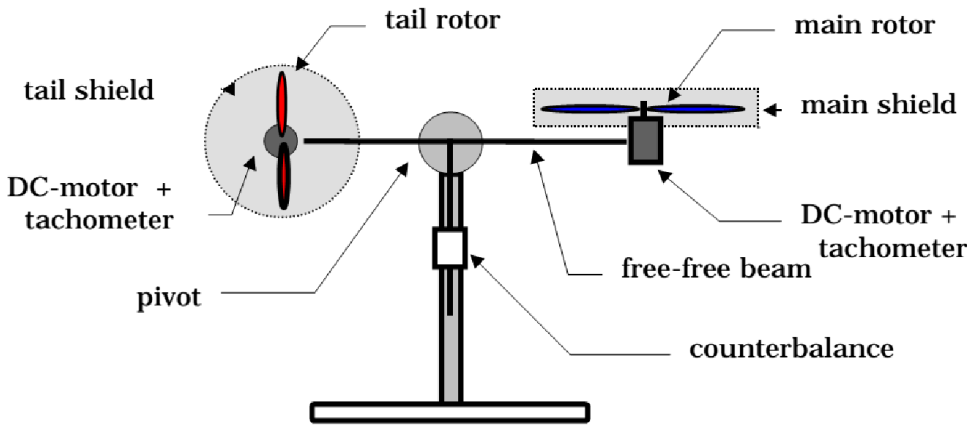

Figure 8: The TRMS [29].

The DC motors are controlled using a DAC card that is installed in a desktop PC. The DAC card is sent commands from the controller that is designed using MATLAB and SIMULINK. Real time control of the TRMS from MATLAB is possible using the real time windows target.

The plant used in the controller design is from [14]. It can be derived by forming the vertical and horizontal equations for momentum, these equations can be seen in Equations (51) and (52).

$$I_1\ddot{\theta}_V = M_1 - M_{FG} - M_{B\theta_V} - M_G \tag{51}$$

$$I_2\ddot{\theta_H} = M_2 - M_{B\theta_H} - M_R \tag{52}$$

Where θ_V and θ_H are the pitch and yaw angles, and M_1 is the nonlinear static characteristic of the rotors and are defined in Equations (53) and (54).

$$M_1 = a_1\tau_1^2 + b_1\tau_1 \tag{53}$$

$$M_2 = a_2\tau_2^2 + b_2\tau_2 \tag{54}$$

$M_{B\theta X}$ are the momentums caused by friction and are defined in Equations (55) and (56).

$$M_{B\theta_V} = B_{1\theta_V}\dot{\theta_V} - \frac{.0326}{2}\sin(2\theta_V)\dot{\theta_V^2} \tag{55}$$

$$M_{B\theta_H} = B_{1\theta_H}\dot{\theta_H} - \frac{.0326}{2}\sin(2\theta_H)\dot{\theta_H^2} \tag{56}$$

M_{FG} is the momentum due to gravity and is defined by Equation (57).

$$M_{FG} = M_g\sin(\theta_V) \tag{57}$$

M_G is the gyroscopic momentum and is defined by Equation (58).

$$M_G = K_{gy}M_1\dot{\theta_H}\cos(\theta_V) \tag{58}$$

M_R is the cross reaction momentum and is given by Equation (59).

$$M_R = \frac{K_C(T_0 s + 1)}{T_P s + 1}M_1 \tag{59}$$

Equation (59) can be rewritten in state space form in Equation (60) and (61). The state ζ is used to describe the cross dynamics of the system.

$$\dot{M}_R = -\frac{1}{T_P}M_R + M_1 \tag{60}$$

$$\zeta = \left(\frac{K_C}{T_P} - \frac{K_C T_P}{T_P^2}\right)M_R + \frac{K_C T_0}{T_P}M_1 \tag{61}$$

The DC motors can be modeled as a first order system, the equations for the vertical and horizontal systems can be seen in Equations (62) and (63).

$$\tau_1 = \frac{K_1}{T_{11}s + T_{10}}U_V \tag{62}$$

$$\tau_2 = \frac{K_2}{T_{21}s + T_{20}}U_H \tag{63}$$

Combining Equations (51) through (63) the system of equations for the plant can be written as (64).

$$\frac{d\theta_V}{dt} = \Omega_V$$

$$\frac{d\Omega_V}{dt} = \frac{1}{I_1}\Big[a_1\tau_1^2 + b_1\tau_1 - M_g\sin(\theta_V) - B_{1\theta_V}\Omega_V + \frac{.0326}{2}\sin(2\theta_V)\,\Omega_V^2 - K_{gy}a_1\cos(\theta_V)\,\Omega_H\tau_1^2 - K_{gy}b_1\cos(\theta_V)\,\Omega_H\tau_1\Big]$$

$$\frac{d\theta_H}{dt} = \Omega_H$$

$$\frac{d\Omega_H}{dt} = \frac{1}{I_2}[a_2\tau_2^2 + b_2\tau_2 - B_{1\theta_H}\Omega_H - \left(\frac{K_C}{T_P} - \frac{K_C T_P}{T_P^2}\right)M_R - \frac{K_C T_0}{T_P}(a_1\tau_1^2 + b_1\tau_1)] \quad (64)$$

$$\dot{M}_R = -\frac{1}{T_P}M_R + a_1\tau_1^2 + b_1\tau_1$$

$$\frac{d\tau_1}{dt} = -\frac{T_{10}}{T_{11}}\tau_1 + \frac{K_1}{T_{11}}U_V$$

$$\frac{d\tau_2}{dt} = -\frac{T_{20}}{T_{21}}\tau_2 + \frac{K_2}{T_{21}}U_H$$

The state variables are defined as (65).

(65)

The plant constants can be seen in Table 1.

Table 1: Parameter Definitions of the TRMS.

Parameter	Description	Value	Units
a_1	Main Rotor Coefficient	.0135	N/A
b_1	Main Rotor Coefficient	.0924	m
a_2	Tail Rotor Coefficient	.01	m
b_2	Tail Rotor Coefficient	.09	m
$B_{1\theta_V}$	Friction Momentum	.003	Nm s/rad
$B_{1\theta_H}$	Friction Momentum	.1	Nm s/rad
M_g	Moment of Gravity	.29	Nm
I_1	Pitch Moment of Inertia	.0535	Kg m^2
I_2	Yaw Moment of Inertia	.02	Kg m^2
K_{gy}	Gyroscopic Momentum	.05	s/rad
T_P	Cross Reaction Momentum Parameter	2	N/A
T_0	Cross Reaction Momentum Parameter	3.5	N/A
K_C	Cross Reaction Momentum Gain	-.2	N/A
T_{10}	Main Rotor Denominator	1	N/A
T_{11}	Main Rotor Denominator	1.1	N/A
T_{20}	Tail Rotor Denominator	1	N/A
T_{21}	Tail Rotor Denominator	1	N/A

Now that the plant model has been derived its stability can be analyzed, this is done in the next section.

3.B. Plant Stability Analysis

It is helpful to look at the way the open-loop system behaves before designing a controller for a plant. The stability of the TRMS can be analyzed by breaking the system into two separate subsystems and then analyzing them independently. Figure 9 and Figure 10 are the phase plane plots for the TRMS system. Figure 9 is for the vertical system and Figure 10 is for the horizontal.

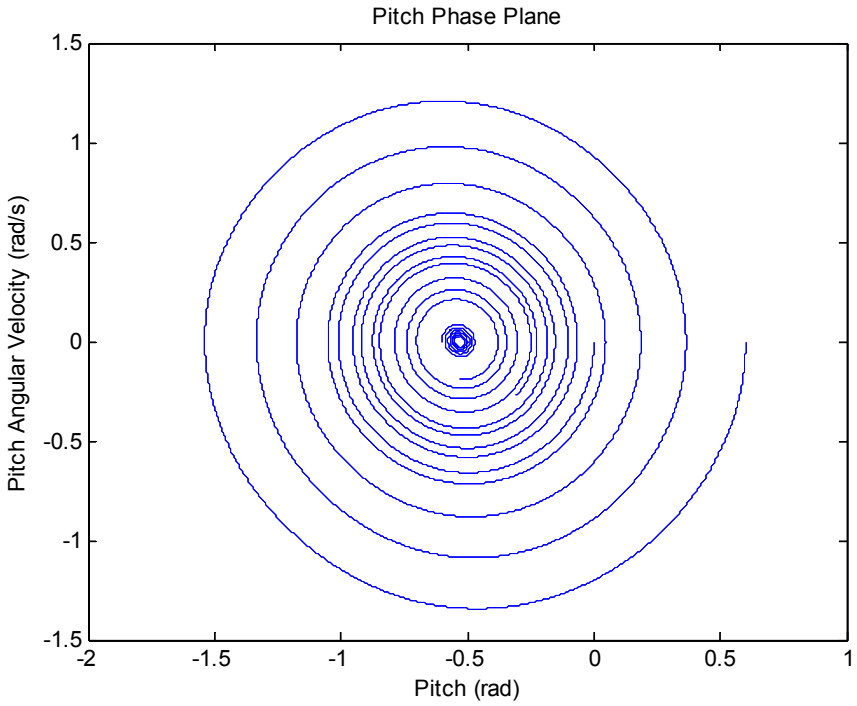

Figure 9: Vertical phase plane of the open-loop TRMS.

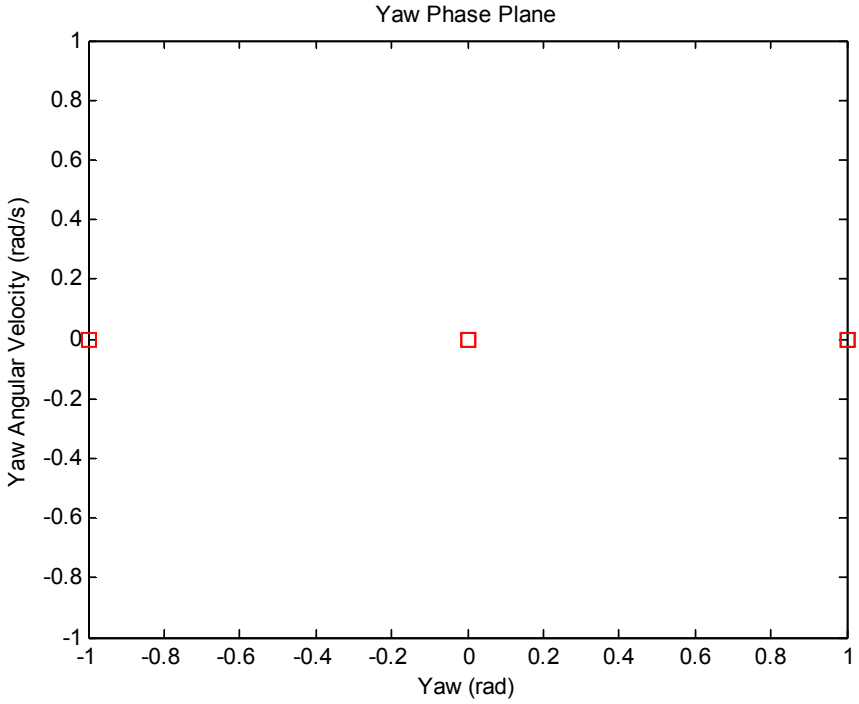

Figure 10: Horizontal phase plane of the open-loop TRMS.

It can be seen from Figure 9 that the vertical system asymptotically approaches the equilibrium point of the system; however it is done slowly and with many oscillations. Figure 10 shows that the horizontal system has no returning force, and because of this when the TRMS moved to a specific yaw it will not move from that point unless an external force is applied.

The slow oscillations of the vertical subsystem and inability of the horizontal subsystem to return to a global equilibrium point are not desirable characteristics for a plant to have and must be compensated for using an external controller.

The next section serves as instructions to install the TRMS and run the simulations using MATLAB and SIMULINK.

3.C. Setting Up the TRMS

To setup the TRMS several things need to be done [30] [31]. First the equipment that is needed is as follows.

1. A PC with Windows XP or Windows 7 32-bit (the drivers would not work with 64 bit)
2. The TRMS
3. PCI-1711-A2 digital to analog converter (DAC) card
4. MATLAB 5.1 and SIMULINK 2.1 at least

The setup procedure must be done as follows.

Step 1: Install the device drivers

1a: Download the 32-bit DLL drivers for the DAC card from the Advantech website.

1b: Run the installer and select “DLL Drivers” installation option.

Step 2: Install the DAC card.

2a: Turn off the computer.

2b: Remove the power cord.

2c: Remove the case cover, and touch the case of the computer to discharge any static.

2d: Insert the card into an available PCI slot, and secure it to the case.

2e: Reassemble the case and turn on the computer.

2f: To verify the installation open device manager on the computer and check for the Advantech PCI card.

Step 3: Configure the DAC card.

3a: Open the start menu and run the "Advantech Device Manager".

3b: Click "add" and then select PCI-1711 from the list, and then click the "install" button.

3c: Click "OK", keep the default settings and click "OK" to finish configuring.

3d: Select the newly installed device and click "Test" to make sure the installation worked. If it did then you will see the "Analog input reading" values changing.

Step 4: Install the SIMULINK files.

4a: Insert the provided Feedback Instruments CD and install the provided SIMULINK models. There are a lot of settings that are required to properly run the models in the real time environment. Those settings are already in place in these models.

4b: Follow the on-screen instructions to complete the installation.

Step 5: Assemble the TRMS (if needed) according to the instructions provided by Feedback Instrument.

Step 6: Connect the AC power cord to the unit. Make sure the ON/OFF switch is off.

Step 7: Connect the digital and analog flat flex cables from the DAC card to the TRMS unit.

7a: Connect the large beige cable from the DAC card to the breakout box provided.

7b: The digital cable is wider than the other two and connects from the bottom connector on the unit to the "Digital I/O" (PL1) connector on the breakout box.

7c: Connect "Analogue Out" (PL3) to the middle connector on the base unit, "CN2"

7d: Connect "Analogue In" (PL2) to the top connector on the base unit, "CN1".

Step 8: Attach the emergency switch to the base station.

Step 9: Now setup MATLAB to use the required C compilers. If this has already been done then this step can be skipped.

9a: Open MATLAB and type "`mex -setup`" into the command line.

9b: Type "`y`", and select a Microsoft Visual Studio C compiler. If the Microsoft Visual Studio Compiler is not visible it must be installed on the computer.

9c: Type "`y`" again and the configuration is complete.

Step 10: To run the models in real time the MATLAB real time windows target (RTWT) needs to be installed. If this is already done then this step can be skipped. To check this type "`rtwho`" into the MATLAB command line.

10a: To do this type "`rtwintgt -install`" into the command line.

10b: Type "`y`".

10c: To make sure the installation worked exit to the desktop and open the "33-949 Twin Rotor MIMO Simulink Models". These should be located on the desktop.

10d: Click "TRMS Simulation Models", then "Model2DOF".

10e: Click "Run" and the simulation should run.

10f: Exit the windows and click the "Up to Parent" button.

10g: Click "TRMS Real-Time Models", then "TRMS2DOF".

10h: Click "Build Model" and wait for it to finish building. If the build was successful it should be indicated in the MATLAB command window.

10i: Turn the base unit ON/OFF switch to "On".

10j: Click "Connect to Target".

10k: Press the green "Start" button on the switch connected to the base unit.

10l: Click "Run". The TRMS should begin to move in a more or less controlled manner determined by the controller. If you wish to end the program either click the "Stop" button or press the "Stop" switch on the base unit.

The TRMS is now setup and ready to be used. The software is not perfect, after being run for extended periods of time it tends to crash. This is because MATLAB runs out of available memory to store the compiled code for the real-time models. When this happens simply restart MATLAB.

Please note that when designing custom controllers or simulations, it is recommended they are created by saving an existing model under a new model name and then building off of the existing working model. This is done because there are a lot of unknown settings that must be configured to properly run the SIMULINK models in real time.

The next section details the design of the system controllers.

4 Controller Design

This section details the design of five controllers. The first is the PID controllers. Then the LQR controller is designed. Finally the three sliding mode controllers are designed. The first two are the first order sliding mode controllers, the QSMC and the SMC with a boundary layer. The final SMC is the SOSMC.

4.A. PID Controller

The PID control structure is shown in Figure 11.

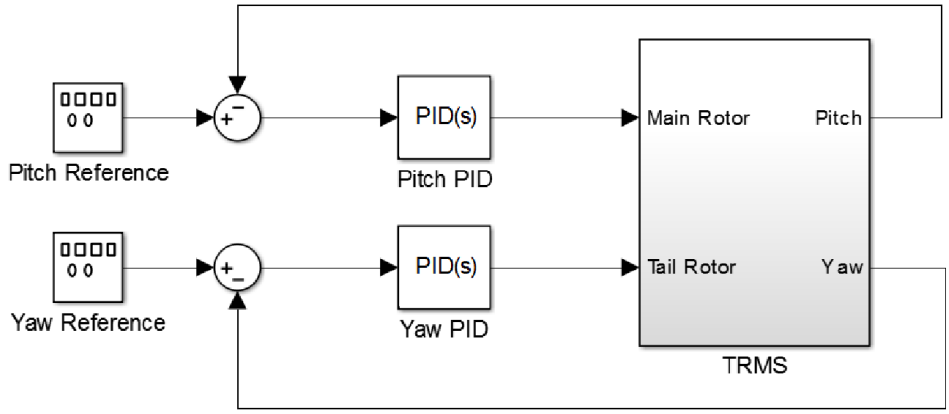

Figure 11: PID control structure.

The PID controllers were tuned using a trial and error method. The TRMS was restricted to 1DOF, then the integral and derivative term were set to 0. The proportional term was increased until the step response began to degrade. Then the integral term was increased until again the step response began to degrade. Finally the process was repeated for the derivative term. Undesirable responses include large overshoot, oscillations and large settling time. This process was repeated for both DOF. Further tuning was performed to achieve a desirable tracking response while maintaining a good step response.

This concludes the design of the PID controllers. The next section details the design of the LQR controller.

4.B. LQR

In this section the LQR controller is designed to include integral action, this is called the LQI controller. First the plant is linearized about the equilibrium point, then the LQR design methodology used earlier is used to create an LQR controller with integral action. Finally the robustness of the controller is investigated at the equilibrium point.

Previous work has been done to find optimal values of the Q and R matrices [12]. However in this thesis the Q and R matrices are weighted manually.

4.B.i. Plant Linearization

The LQR solution presented here requires a linear system. The nonlinear system presented in Equation (64) is linearized using Equations (66) and (67).

$$A = \left.\frac{dF}{dx}\right|_0 \tag{66}$$

$$B = \left.\frac{dF}{du}\right|_0 \tag{67}$$

Applying Equations (66) and (67) to the Equation (64), Equations (68) and (69) are found. These are the linear model of the plant.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{68}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{K_1}{T_{11}} & 0 \\ 0 & \frac{K_2}{T_{21}} \end{bmatrix} \tag{69}$$

In Equation (68) the *A* matrix is represented in block form for readability. The A_{11}, A_{12}, A_{21}, and A_{22} matrices are shown in Equations (70) through (73).

$$A_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{M_g}{I_1} & -\frac{B_{1\theta_V}}{I_1} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{B_{1\theta_H}}{I_2} \end{bmatrix} \tag{70}$$

$$A_{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{2a_1 + b_1}{I_1} & 0 \\ 0 & 0 & 0 \\ \frac{K_C(T_0 + T_P)}{T_P^2} & \frac{K_C T_0(2a_1 + b_1)}{I_2 T_P} & \frac{2a_2 + b_2}{I_2} \end{bmatrix} \tag{71}$$

$$A_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{72}$$

$$A_{22} = \begin{bmatrix} -\frac{1}{T_P} & 2a_1 + b_1 & 0 \\ 0 & -\frac{T_{10}}{T_{11}} & 0 \\ 0 & 0 & -\frac{T_{20}}{T_{21}} \end{bmatrix} \tag{73}$$

It can be seen that there is already a pole at the origin. However this pole is a result of the linearization, and the location of the poles will change with the system state. The guarantee that there will be zero steady state error in a step input integral action will be imparted onto the suboptimal loop.

4.B.ii. Controller Design

The LQI controller was designed by varying the Q and R matrices. Greater weight was given to the error states than the other states because a tracking controller was desired. Many simulations were performed in order to achieve satisfactory performance. Once that was done the robustness was checked.

To check the gain and phase margins of the closed loop system, it was linearized at the origin. Bode plots of the transfer function matrix were plotted and the margins were found. The gain margin for the controller shown here is more than 50dB, and an infinite phase margin. This can be seen in Figure 12.

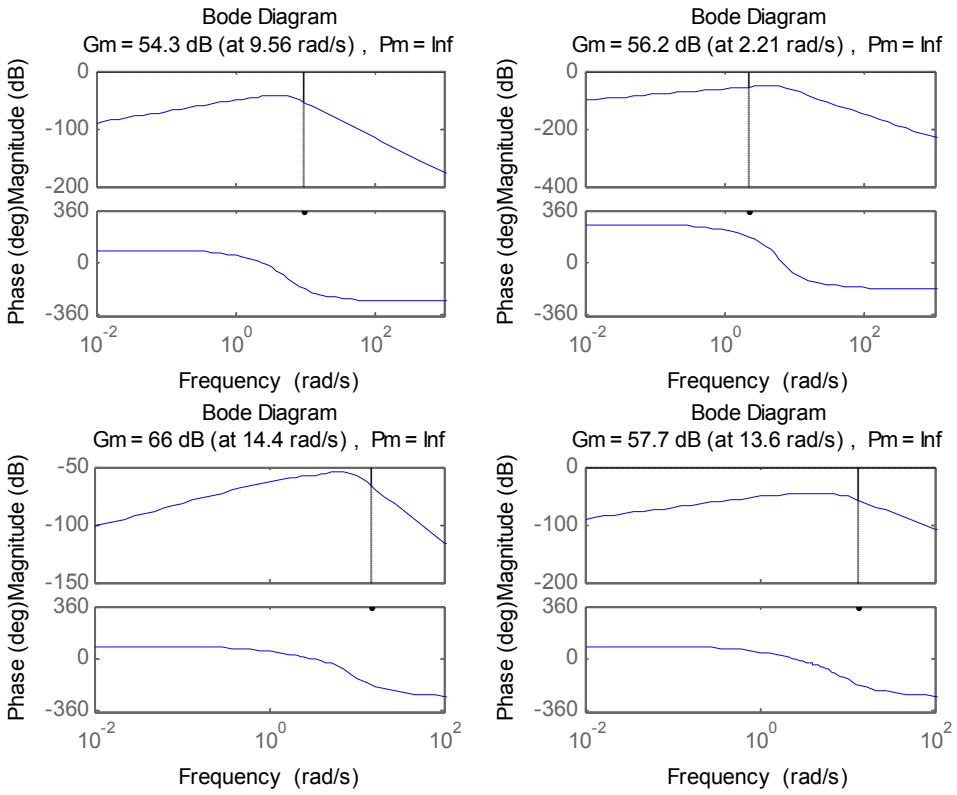

Figure 12: Closed loop bode plots of the TRMS using the suboptimal LQI controller.

This frequency analysis is only valid about the linearization point. This is where the robustness of the LQI controller becomes important. It is important the closed loop system remains stable as the system state varies. The fact that the gain and phase margins are so high indicate a robust controller.

The LQI control design is complete, the next section details the design of the SMCs.

4.C. SMC

Three SMCs are designed in this Thesis. The first two, the QSMC and the SMC with a boundary layer (SMCB). This next section details the process to decouple the systems into tow SISO subsystems, the design of the SMCs, the design of the adaptation law, and an analysis of the controllers stability.

4.C.i. Plant Decomposition

The model shown in Equation (64) is underactuated, that is it has fewer inputs than states, and is in a form that is ill-suited for SMC design. To apply ideal SMC design, the system must be square so that the B matrix is invertible. To get around this, the system will be decomposed into a vertical and horizontal systems. The systems are first separated, into two SISO systems then linearized about the equilibrium point $X_E = [0\ 0\ 0\ 0\ 0\ 0]^T$ [10]. Equations (74) and (75) represent the decomposed system.

$$\dot{X_V} = A_V X_V + B_V U_V + \Delta F_V \tag{74}$$

$$\dot{X_H} = A_H X_H + B_H U_H + \Delta F_H \tag{75}$$

Where $X_V = [X_1\ X_3\ X_5]^T$ and $X_H = [X_2\ X_4\ X_6]^T$. The uncertainty blocks, ΔF_V and ΔF_H are the known disturbances created by decoupling the systems, they are defined by Equation (76).

$$\Delta F_V = \begin{bmatrix} 0 \\ \Delta f_{V2} \\ 0 \end{bmatrix} \qquad \Delta F_H = \begin{bmatrix} 0 \\ \Delta f_{H2} \\ 0 \end{bmatrix} \tag{76}$$

There is a problem with using Equation (64) because there is a state that represents the cross coupling between the pitch and yaw subsystems. It is ambiguous whether this state should belong to the pitch or yaw subsystem. In order to eliminate this ambiguous state from the system the M_R state is approximated as a 0th order system. Equation (59) is approximated by Equation (77).

$$M_R = \frac{K_C T_0}{T_P} M_1 \tag{77}$$

This eliminates the cross-coupled state, and the new plant is shown in Equation (78).

$$\begin{gathered} \frac{d\theta_V}{dt} = \Omega_V \\ \frac{d\Omega_V}{dt} = \frac{1}{I_1}[a_1\tau_1^2 + b_1\tau_1 - M_g \sin(\theta_V) - B_{1\theta_V}\Omega_V + \frac{.0326}{2}\sin(2\theta_V)\,\Omega_V^2 \\ -K_{gy}a_1 \cos(\theta_V)\,\Omega_H\tau_1^2 - K_{gy}b_1\cos(\theta_V)\Omega_H\tau_1] \\ \frac{d\theta_H}{dt} = \Omega_H \\ \frac{d\Omega_H}{dt} = \frac{1}{I_2}[a_2\tau_2^2 + b_2\tau_2 - B_{1\theta_H}\Omega_H - 1.75K_c a_1\tau_1^2 - 1.75K_c b_1\tau_1] \end{gathered} \tag{78}$$

$$\frac{d\tau_1}{dt} = -\frac{T_{10}}{T_{11}}\tau_1 + \frac{K_1}{T_{11}}U_V$$
$$\frac{d\tau_2}{dt} = -\frac{T_{20}}{T_{21}}\tau_2 + \frac{K_2}{T_{21}}U_H$$

The A_V, A_H, B_V, and B_H matrices are found by reducing (78) into two different systems of equations defined by the state vectors X_V and X_H, then linearizing them about the equilibrium points, $X_{EV} = [0\ 0\ 0]^T$ and $X_{EH} = [0\ 0\ 0]^T$. The result is in Equations (79) and (80)

$$A_V = \left.\frac{dF(X_V)}{dX_V}\right|_{X_{EV}} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{M_g}{I_1} & -\frac{B_{1\theta_V}}{I_1} & \frac{b_1}{I_1} \\ 0 & 0 & -\frac{T_{10}}{T_{11}} \end{bmatrix}$$
$$A_H = \left.\frac{dF(X_H)}{dX_H}\right|_{X_{EH}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{B_{1\theta_H}}{I_2} & \frac{b_2}{I_2} \\ 0 & 0 & -\frac{T_{20}}{T_{21}} \end{bmatrix} \tag{79}$$

$$B_V = \left.\frac{dF(X_V)}{dU_V}\right|_{X_{EV}} = \begin{bmatrix} 0 \\ 0 \\ \frac{K_1}{T_{11}} \end{bmatrix}$$
$$B_H = \left.\frac{dF(X_H)}{dU_H}\right|_{X_{EH}} = \begin{bmatrix} 0 \\ 0 \\ \frac{K_2}{T_{21}} \end{bmatrix} \tag{80}$$

Comparing Equations (74) through (76) and Equation (78), to Equation (79) and (80), the resulting values of ΔF_V and ΔF_H can be found. They are shown in Equation (81) and Equation (82)

$$\Delta f_{V2} = \frac{1}{I_1}[a_1 x_5^2 + b_1 x_5 - M_g \sin(x_1) - B_{1\theta_V} x_3 - K_{gy} b_1 \cos(x_1) x_4 x_5] + \frac{M_g}{I_1} x_1 + \frac{B_{1\theta_V}}{I_1} x_3 - \frac{b_1}{I_1} x_5 + \frac{.0326}{2} \sin(2x_1)\, x_4^2 - K_{gy} a_1 \cos(x_1)\, x_4 x_5^2 \tag{81}$$

$$\Delta f_H = \frac{1}{I_2}\left[a_2 \tau_2^2 + b_2 \tau_2 - B_{1\theta_H}\Omega_H - 1.75 K_c a_1 \tau_1^2 - 1.75 K_c b_1 \tau_1\right] + \frac{B_{1\theta_H}}{I_2} x_4 - \frac{b_2}{I_2} x_6 \tag{82}$$

To make the notation simpler when deriving the SMC, Equations (74) and (75) can be re-written in block form as (83) and (84).

$$\begin{bmatrix} \dot{X_{V1}} \\ \dot{X}_{V2} \end{bmatrix} = \begin{bmatrix} A_{V11} & A_{V12} \\ A_{V21} & A_{V22} \end{bmatrix} \begin{bmatrix} X_{V1} \\ X_{V2} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{V2} \end{bmatrix} U_V + \begin{bmatrix} \Delta F_{V1} \\ 0 \end{bmatrix} \tag{83}$$

$$\begin{bmatrix} \dot{X_{H1}} \\ \dot{X}_{H2} \end{bmatrix} = \begin{bmatrix} A_{H11} & A_{H12} \\ A_{H21} & A_{H22} \end{bmatrix} \begin{bmatrix} X_{H1} \\ X_{H2} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{H2} \end{bmatrix} U_V + \begin{bmatrix} \Delta F_{H1} \\ 0 \end{bmatrix} \tag{84}$$

Where the dimensions of the matrices are $A_{V11} \in \Re^{2x2}$, $A_{V21} \in \Re^{1x2}$, $A_{V12} \in \Re^{2x1}$, $A_{V22} \in \Re^{1x1}$, $A_{H11} \in \Re^{2x2}$, $A_{H21} \in \Re^{1x2}$, $A_{H12} \in \Re^{2x1}$, $A_{H22} \in \Re^{1x1}$, $B_{H2} \in \Re^{1x1}$ and $B_{V2} \in \Re^{1x1}$.

The state vectors are partitioned so the sub-matrices are $X_{V1} \in \Re^{2x1}$, $X_{V2} \in \Re^{1x1}$, $X_{H1} \in \Re^{2x1}$, and $X_{H2} \in \Re^{1x1}$.

The error vectors are defined in Equations (85) and (86). The R_V and R_H matrices are the reference vectors defined as $R = [r_{x1}\ r_{x2}]^T$, where $r_{x1} = [r_x\ \dot{r_x}]^T$ and $r_{x1} = [0]$.

$$E_V = \begin{bmatrix} e_{V1} \\ e_{V2} \end{bmatrix} = \begin{bmatrix} X_{V1} \\ X_{V2} \end{bmatrix} - \begin{bmatrix} r_{V1} \\ r_{V2} \end{bmatrix} \tag{85}$$

$$E_H = \begin{bmatrix} e_{H1} \\ e_{H2} \end{bmatrix} = \begin{bmatrix} X_{H1} \\ X_{H2} \end{bmatrix} - \begin{bmatrix} r_{H1} \\ r_{H2} \end{bmatrix} \tag{86}$$

Replacing X_x with E_x in (83) and (84), the error state equation is found to be (87) and (88).

$$\dot{E}_V = A_V E_V + B_V U_V + \Delta F_V \tag{87}$$

$$\dot{E}_H = A_H E_H + B_H U_H + \Delta F_H \tag{88}$$

4.C.ii. Controller Design

The first step for the controller design is to define the sliding surface. In this case the same sliding surface is used for both the vertical and the horizontal sliding surface. The proportional Integral (PI) sliding surface is defined in Equation (89).

$$S = {T_{1x}}^T E_x + T_{2x}^T \int_0^t E_x dr \tag{89}$$

Where $T_{1x} = [T_{11x} T_{12x}]^T$ and $T_{2x} = [T_{21x} T_{22x}]^T$, where $T_{11x} \in \Re^{1x2}$, $T_{21x} \in \Re^{1x1}$, $T_{12x} \in \Re^{1x2}$, $T_{x22} \in \Re^{1x1}$ and all the elements of T_{1x} and T_{2x} are greater than or equal to 0.

The controller design for the horizontal is found by taking the derivative of Equation (89). The disturbance term is not included because it cannot be predicted perfectly by the controller. The system must be written this way to guarantee that the system of equations can be solved for the input.

$$\dot{S_H} = T_{1H}A_H E_H + T_{2H}^T E_H + T_{1H}B_H U_H \tag{90}$$

The reaching law is defined in Equation (91). Here T_{1H} is a positive constant, and $sign()$ is the sign function.

$$\dot{S}_H = -K_H sign(S_H) \tag{91}$$

The control law for the horizontal subsystem is found by equating Equations (90) and (91).

$$U_H = -(T_{1H}B_H)^{-1}[T_{1H}A_H E_H + T_{2H}^T E_H + K_H sign(S_H)] \tag{92}$$

There are more constraints placed upon the values chosen for C_H and D_H than they simply must be positive. Based on the work done by [10] the real part of the eigenvalues of the matrix $A_{H11} - A_{H12}T_{12H}^{-1}T_{11H}$ must be negative.

Similarly for the vertical subsystem, Equation (90) can be re-written.

$$\dot{S_V} = T_{1V}A_V E_V + T_{2V}^T E_V + T_{1V}B_V U_V \tag{93}$$

The same reaching law is used for the vertical subsystem, it can be seen in Equation (94). Again T_V is a positive constant.

$$\dot{S}_V = -K_V sign(S_V) \tag{94}$$

Equating Equation (93) and Equation (94) the vertical control law is found to be Equation (95).

$$U_V = -(T_{1V}B_V)^{-1}[T_{1V}A_V E_V + T_{2V}^T E_V + K_V sign(S_V)] \tag{95}$$

Again the real part of the eigenvalues of the matrix $A_{V11} - A_{V12}T_{12V}^{-1}T_{11V}$ must be negative.

The controller design is done, however that stability of the closed loop system must be proven. This is done in the next section.

4.C.iii. Controller Stability

Stability of the closed loop system is found by defining the Lyapunov Function as Equation (96).

$$V = \frac{1}{2}S^2 > 0 \tag{96}$$

Taking the derivative of Equation (96) yields Equation (97).

$$\dot{V} = S\dot{S} < 0 \tag{97}$$

For the horizontal subsystem, using Equation (90) with Equation (97) yields Equation (98).

$$\dot{V} = S_H[T_{1H}A_H E_H + T_{2H}^T E_H + T_{1H}B_H U_H + \Delta F_V] \tag{98}$$

Using Equation (92) with Equation (98) yields Equation (99).

$$\begin{gathered}\dot{V} = S_H\{\Delta F_H - K_H sign(S_H)\} < 0 \\ S_H \Delta F_H < K_H |S_H|\end{gathered} \tag{99}$$

So if Equation (99) is true then the system will be stable. A similar analysis can be done for the vertical subsystem.

$$\dot{V} = S_V[T_{1V}A_V E_V + T_{2V}^T E_V + T_{1V}B_V U_V + \Delta F_V] < 0 \tag{100}$$

Using Equation (95) with Equation (100) yields (101).

$$\begin{gathered}\dot{V} = S_V[\Delta F_V - K_V sign(S_V)] < 0 \\ S_V \Delta F_V < K_V |S_V|\end{gathered} \tag{101}$$

If (101) is true then the system is stable.

In this case the magnitude of ΔF_V and ΔF_H can be calculated. However in reality the magnitude may not be known. There could be unmodeled dynamics, parameter variations or environmental disturbances that were not taken into account. This can be compensated for in one of two ways. The first is to make K_X arbitrarily large, however this can result in an unnecessarily large control effort, it can also make the chattering that is characteristic of SMC very difficult to attenuate. The other way is to define an adaptive control law to find an optimal value of K_X over time. This is done in the next section.

4.C.iv. Adaptive Control Law

The adaptive control law presented here is derived from the work done by Mondal et al [10]. Consider the Lyapunov Candidate in Equation (102).

$$V = \frac{1}{2}S^2 + \frac{\gamma}{2}\tilde{K}^2 \tag{102}$$

Where

$$\tilde{K} = \hat{K} - K \tag{103}$$

And γ is a positive constant, and $\hat{K}$ is the estimated value of the actual gain, K. Taking the derivative of Equation (102).

$$\dot{V} = S\dot{S} + \gamma\tilde{K}\dot{\tilde{K}} < 0 \tag{104}$$

For the horizontal subsystem, using Equation (90) and Equation (103) with Equation (104).

$$\dot{V} = S_H[T_{1H}A_H E_H + T_{2H}^T E_H + T_{1H}B_H U_H] + \gamma_H \tilde{K}_H \dot{\hat{K}}_H < 0 \tag{105}$$

Using Equation (92) with Equation (105) yields Equation (106).

$$\dot{V} = S_H\left[\Delta F_H - \hat{K}_H sign(S_H)\right] + \gamma_H \tilde{K}_H \dot{\hat{K}}_H = \Delta F_{H1} S_H - \hat{K}_H|S_H| + \gamma_H(\hat{K}_H - K_H)\dot{\hat{K}}_H < 0 \tag{106}$$

If $\dot{\hat{T}}_H$ is defined as Equation (107).

$$\dot{\hat{K}}_H = \frac{1}{\gamma_H}|S_H| \tag{107}$$

Then using Equation (107) with Equation (106) yields Equation (107).

$$\dot{V} = \Delta F_H S_H - K_H|S_H| < 0$$
$$\Delta F_{H1} S_H < K_H|S_H| \tag{108}$$

Equation (108) is true as long as $K_H > \Delta F_H$ and the adaptation law used is Equation (107). A similar analysis can be done for the vertical system. Using Equation (93) and Equation (103) with Equation (102) yields Equation (109).

$$\dot{V} = S_V[T_{1V}A_V E_V + T_{2V}^T E_V + T_{1V}B_V U_V] + \gamma_V \tilde{T}_V \dot{\hat{T}}_V < 0 \tag{109}$$

Using Equation (95) with Equation (109) yields Equation (110).

$$\dot{V} = S_V[\Delta F_V - \hat{K}_V sign(S_V)] + \gamma_V(\hat{K}_V - K_V)\dot{\tilde{K}}_V < 0 \tag{110}$$

Again defining the adaptive control law as Equation (111).

$$\dot{\hat{K}}_V = \frac{1}{\gamma_V}|S_V| \tag{111}$$

Combining Equation (110) and Equation (111) yields Equation (112).

$$\begin{aligned} \dot{V} = \Delta F_V S_V - K_V|S_V| < 0 \\ \Delta F_V S_V < K_V|S_V| \end{aligned} \tag{112}$$

Equation (112) is true if $K_V > \Delta F_V$ and the adaptation law Equation (111) is used.

In reality the sliding surface and its derivative will never converge to zero at the same time. Because of this, K_V and K_H will continue to grow over time. To compensate for this, a dead zone is used. The sliding surface will oscillate around zero slightly over time. The dead zone makes sure this would not cause the switching gain to grow uncontrollably. The dead zone is defined by Equation (113).

$$S = \begin{cases} 0, |S| < \varepsilon \\ S, |S| > \varepsilon \end{cases} \tag{113}$$

This completes the design of the first order SMCs. In the next section the second order SMC is designed.

4.D. SOSMC

The process for designing the SOSMC is slightly different from that of the first order SMC. This is because the SOSMC is derived from the second derivative of the sliding surface instead of the first. For the horizontal subsystem, consider Equation (89) from the previous section.

$$S = T_{1x}E_x + T_{2H}\int_0^t E_x dr$$

$$S_H = T_{1H}E_H + T_{2H}\int_0^t E_H dr$$

And it's first derivative, Equation (90) from the previous section.

$$\dot{S_H} = T_{1H}(A_H E_H + B_H U_H + \Delta F_H) + T_{2H}E_H$$

Now taking its derivative again yields Equation (114).

$$\begin{aligned} \ddot{S}_H = (T_{1H}A_H^2 + T_{2H}A_H)E_H + (T_{1H}A_H + T_{2H})B_H U_H + T_{1H}\left(A_H \Delta F_H + \Delta \dot{F}_H\right) \\ + T_{1H}B_H \dot{U}_H \end{aligned} \tag{114}$$

If we define the discontinuous derivative of the input as $\dot{U}_H$ that will stabilize the system, then the actual input U_H will be continuous. The controller cannot be derived using the Lyapunov method from before because it requires the second derivative of the sliding surface to be present, which it would not be. To get around this we define a sliding function as Equation (115).

$$\sigma = \dot{S} + \alpha S \tag{115}$$

Taking the derivative of Equation (115) yields Equation (116).

$$\dot{\sigma} = \ddot{S} + \alpha \dot{S} \tag{116}$$

So for the horizontal subsystem Equation (116) becomes Equation (117).

$$\begin{aligned} \dot{\sigma}_H = {} & (T_{1H}A_H^2 + T_{2H}A_H)E_H + (T_{1H}A_H + T_{2H})B_H U_H + T_{1H}\left(A_H \Delta F_H + \Delta \dot{F}_H\right) \\ & + T_{1H}B_H \dot{U}_H + \alpha_H T_{1H}(A_H E_H + B_H U_H + \Delta F_H) + \alpha_H T_{2H}E_H \end{aligned} \tag{117}$$

The same reaching law as the SMCB and QSMC designs in the previous section is used.

$$\dot{\sigma}_H = -K_H sign(S_H)$$

Equating Equation (116) with the reaching law the control law is derived as Equation (118).

$$\begin{aligned} \dot{U}_H = {} & -(T_{1H}B_H)^{-1}[(T_{1H}A_H^2 + T_{2H}A_H + \alpha_H T_{2H} + \alpha_H T_{1H}A_H)E_H \\ & + (T_{1H}A_H + T_{2H} + \alpha_H T_{1H})B_H U_H + T_{1H}\left(A_H \Delta F_H + \Delta \dot{F}_H + \alpha_H T_{1H}\Delta F_H\right) \\ & + K_H sign(S_H)] \end{aligned} \tag{118}$$

For the vertical subsystem the sliding surface is defined as Equation (89), and its derivative is defined as Equation (93) in the previous section. It is rewritten below for convenience.

$$\dot{S_V} = T_{1V}A_V E_V + T_{2V}^T E_V + T_{1V}B_V U_V$$

Taking the derivative of the vertical sliding surface again the following is found.

$$\ddot{S}_V = (T_{1V}A_V^2 + T_{2V}A_V)E_V + (T_{1V}A_{VA} + T_{2V})B_V U_V + T_{1V}\left(A_V \Delta F_V + \Delta \dot{F}_V\right) + T_{1V}B_V \dot{U}_V \tag{119}$$

Using the same sliding function derived in Equation (115) and (116).

$$\begin{aligned} \dot{\sigma}_V = {} & (T_{1V}A_V^2 + T_{1V}A_V)E_V + (T_{1V}A_V + T_{2V})B_V U_V + T_{1V}\left(A_V \Delta F_V + \Delta \dot{F}_V\right) + T_{1V}B_V \dot{U}_V \\ & + \alpha_V T_{1V}(A_V E_V + B_V U_V + \Delta F_V) + \alpha_V T_{2V}E_V \end{aligned} \tag{120}$$

Now using the same reaching law as before the control law is defined as Equation (121).

$$\begin{aligned}\dot{U}_V = -(T_{1V}B_V)^{-1}[&(T_{1V}A_V^2 + T_{2V}A_V + \alpha_V T_{2V} + \alpha_V T_{1V}A_V)E_V \\ &+ (T_{1V}A_V + T_{2V} + \alpha_V T_{1V})B_V U_V + T_{1V}\left(A_V \Delta F_V + \Delta \dot{F}_V + \alpha_V T_{1V}\Delta F_V\right) \\ &+ K_V sign(S_V)]\end{aligned} \tag{121}$$

There is an expression for the unknown term ΔF_V and ΔF_H, however there is none for their derivatives. They can be derived by taking the time derivative of Equation (81) and Equation (82), but this is cumbersome. A simpler solution is to use the method used when deriving the first order SMC, which is to simply ignore the terms and let the adaptive switching gain take care of it. Using this Equation (118) and (121) become Equation (122) and (123).

$$\begin{aligned}\dot{U}_H = -(T_{1H}B_H)^{-1}[&(T_{1H}A_H^2 + T_{2H}A_H + \alpha_H T_{2H} + \alpha_H T_{1H}A_H)E_H \\ &+ (T_{1H}A_H + T_{2H} + \alpha_H T_{1H})B_H U_H + K_H sign(S_H)]\end{aligned} \tag{122}$$

$$\begin{aligned}\dot{U}_V = -(T_{1V}B_V)^{-1}[&(T_{1V}A_V^2 + T_{2V}A_V + \alpha_V T_{2V} + \alpha_V T_{1V}A_V)E_V \\ &+ (T_{1V}A_V + T_{2V} + \alpha_V T_{1V})B_V U_V + K_V sign(S_V)]\end{aligned} \tag{123}$$

The controller design is complete. The next step is to design the observer to provide full state feedback. The Luenberger Observer is derived in the next section.

4.E.Luenberger Observer

There are many options available to provide estimates of the system states. There has been work done in the area of nonlinear observers. The Extended Kalman Filter is capable of handling some nonlinearity, the Unscented Kalman Filter does a better job but is very complicated to implement. In addition Kalman Filters need to be reset ever so often to keep them from going unstable [32]. Sliding Mode Observers have also been implemented but again they are even more complicated systems. More recently there has been work done to extend the Luenberger observer to nonlinear systems [14]. However in this application a simple Luenberger observer proved sufficient.

A Luenberger observer was implemented to provide full state feedback for the controllers. Because the plant is nonlinear a LQR design process was used to obtain the feedback matrix L. This was done so the Q and R matrices of the LQR design methodology can be chosen to reduce error from specific states as needed. For the LQI controller the Q and R matrices of the observer were chosen as Equation (124)

$$Q = diag([100; 100; 10000000; 10000000; 1000000; 1; 1])$$
$$R = diag([1; 1]) \tag{124}$$

For observer for the SMC were chosen as Equation (125).

$$Q = diag([100; 100000000; 1; 100; 1000000000; 1])$$
$$R = diag([1; 1]) \tag{125}$$

The design of the system is done. The next section shows the results of the simulations.

5 Results

This section displays and analyzes the results of the controllers both through simulation and experimentally. The first section shows the simulation results, the second shows the experimental results. Both sections include both step responses and tracking signals. In addition to be compared against each other they are also against those in the literature.

5.A. Simulation Results

There are three subsections in the simulation results. The first is the step response of the controllers, the second is the tracking results, finally the simulations are compared against other from the literature.

5.A.i.Step Response

The LQI controller as well as the SMCs were implemented in hardware using the Twin Rotor MIMO system supplied by Feedback Instruments. The simulations were done using MATLAB and SIMULINK and a sample time of "1ms" and the "ode5" solver was used. A Luenberger observer was used to provide the full state feedback.

Two simulations were performed for each controller. The first is with a step reference signal. The desired pitch was .2π radians and the desired yaw was .6π radians. The second was a tracking reference signal with an amplitude of .2π radians and a frequency of .025Hz.

Equations (126) and (127) show the values of the Q and R matrices used for the LQI controller.

$$Q = diag([10^4; 10^4; 10^3; 10^3; 1; 1; 1; 10^4; 10^4]) \tag{126}$$

$$R = I \tag{127}$$

For the SMC using a boundary layer (BSMC) the parameters for the horizontal controller are defined as $C_H = [40, 10, 20]^T$, $D_H = [10\ 50\ 0]^T$, $\varphi_H = .1$, $\gamma_H = 2$, $\varepsilon_H = 3$, and the initial value of the switching gain is set to 3. The controller parameters for the vertical system are defined as $C_V = [22, 5, 6]^T$, $D_V = [20\ 2.5\ 0]^T$, $\varphi_V = .5$, $\gamma_V = 2$, $\varepsilon_V = 3$, and the initial value of the switching gain is set to 10.

The QSMC parameters for the horizontal controller are defined as $C_H = [40, 6, 8]^T$, $D_H = [10\ 20\ 0]^T$, $\varphi_H = .5$, $\gamma_H = 2$, $\varepsilon_H = 3$, and the initial value of the switching gain is set to 3. The controller parameters for the vertical system are defined as $C_V = [25, 11, 5]^T$, $D_V = [25\ 5\ 0]^T$, $\varphi_V = .5$, $\gamma_V = 2$, $\varepsilon_V = 2$, and the initial value of the switching gain is set to 35.

The SOSMC parameters for the horizontal controller are defined as $C_H = [40\ 6\ \ 8]^T$, $D_H = [25\ 20\ 0]^T$, $\gamma_H = 2.5$, $\varepsilon_H = 7$, $\alpha_H = 10$, $K_H = 1$ and the initial value of the switching gain is set to 10. The controller parameters for the vertical system are defined as $C_V = [30\ 13\ 5]^T$, $D_V = [60\ 15\ 0]^T$, $\gamma_V = 2.5$, $\varepsilon_V = 7$, , $\alpha_V = 15$, $K_V = 4$ and the initial value of the switching gain is set to 10.

Figure 13 shows the simulated step response of the systems. The SOSMC perform the best, providing fast rise time with minimal overshoot and ringing. The LQI and QSMC provides excellent performance as well. The PID controller's performance was the worst with the most overshoot in the case of the yaw and with a slow rise time and a lot of ringing for the pitch.

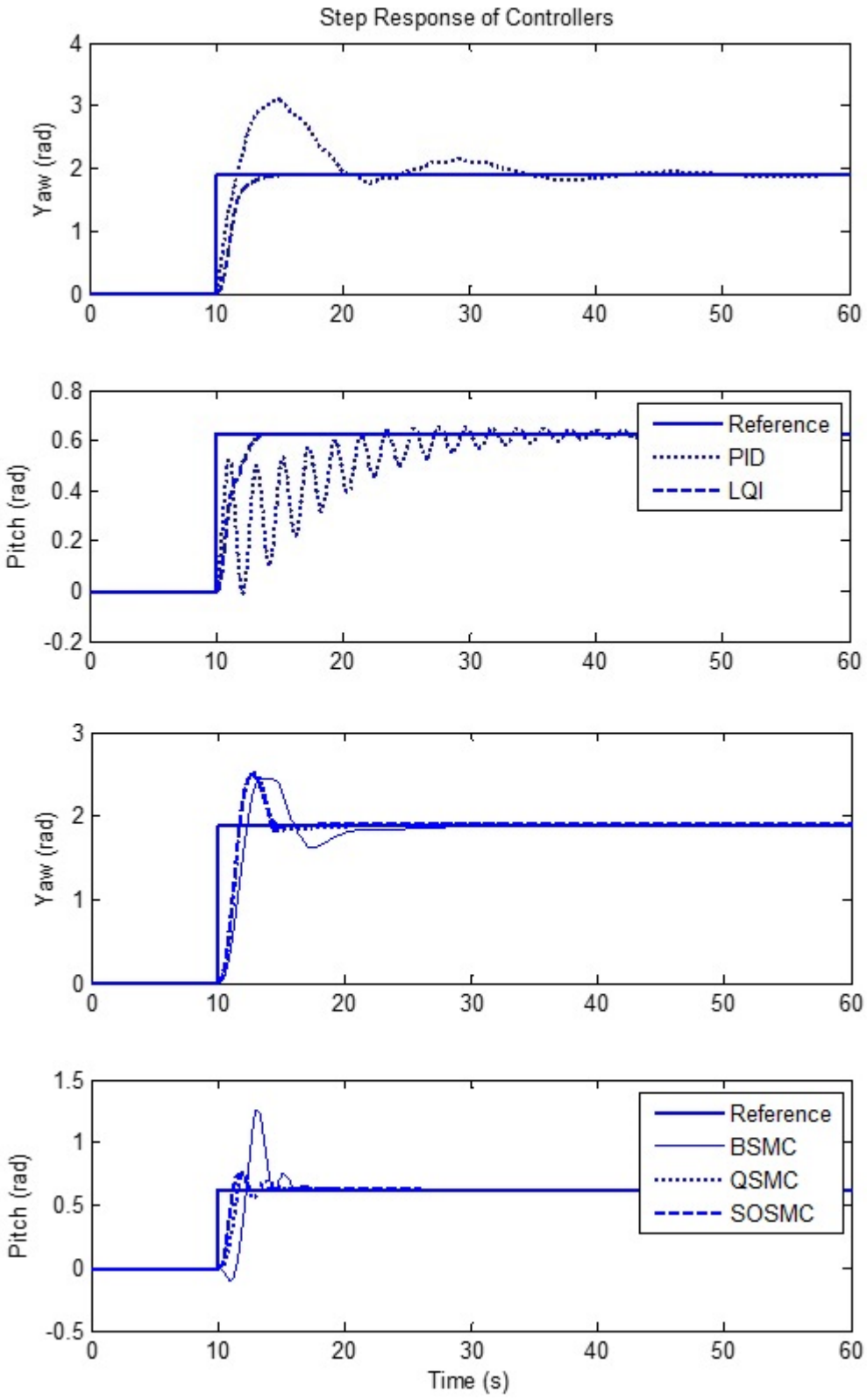

Figure 13: Comparison of simulated step responses.

Figure 14 shows the switching gains for the three SMCs. The switching gains are very large requiring the chatter suppression techniques to perform very well.

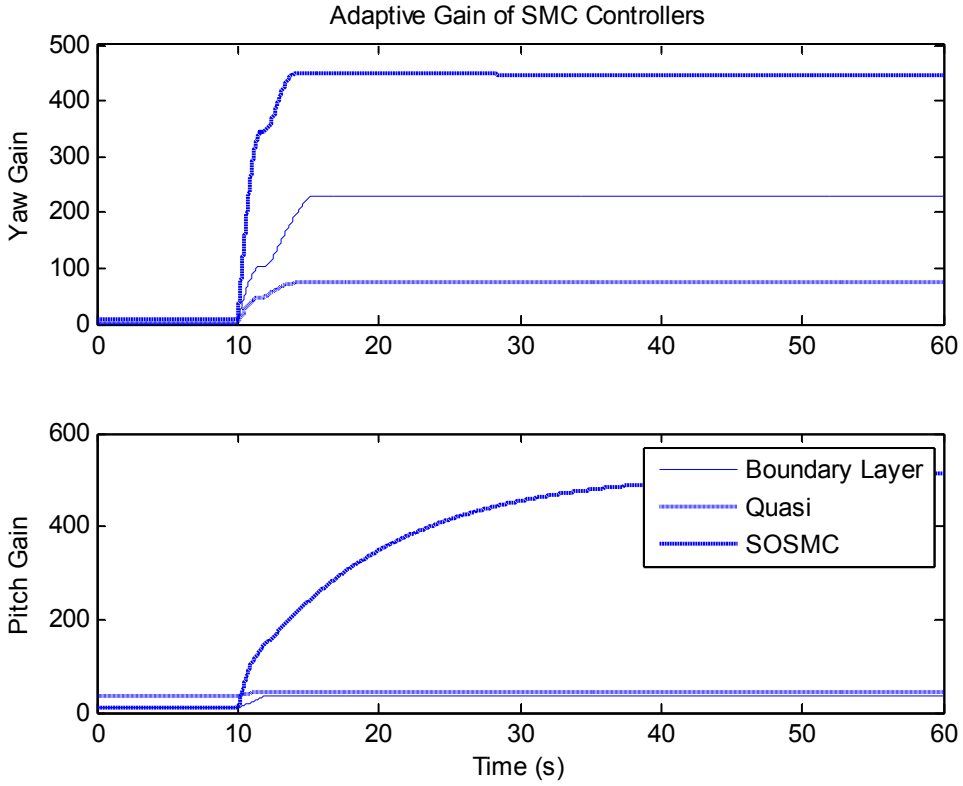

Figure 14: Comparison of simulated switching gains.

Figure 15 shows the control effort for the three SMCs. All three controllers are free of chattering. The saturation of the control effort in Figure 15 is due to the limits of the systems control input.

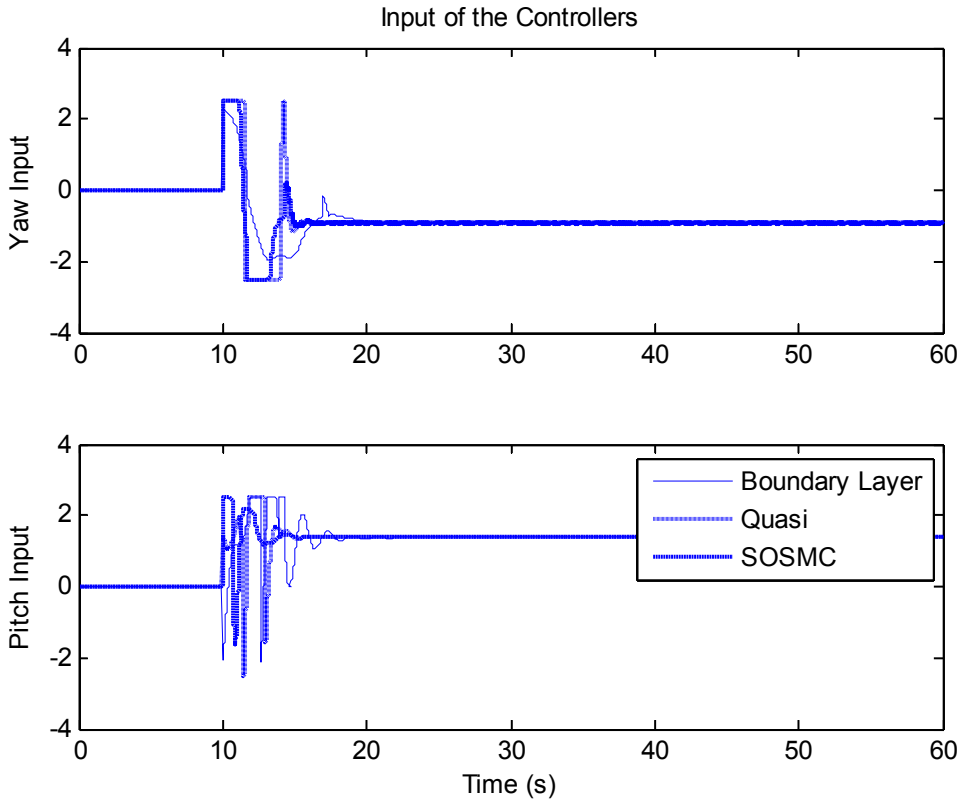

Figure 15: Comparison of simulated control effort of SMCs.

5.A.ii. Tracking Response

The controllers were also simulated using a tracking reference signal as specified earlier. Figure 16 shows the simulated tracking response of the controllers. Clearly the PID controller does not provide good tracking. Further insight into the performance can be had by inspecting Figure 17 which shows the simulated tracking error of the system.

Again the PID controller provides the worst tracking. Despite the excellent step response of the LQI controller the tracking response is as good. The QSMC and SOSMC provide the best tracking performance. Table 2 shows the mean squared error (MSE) of the simulated controllers.

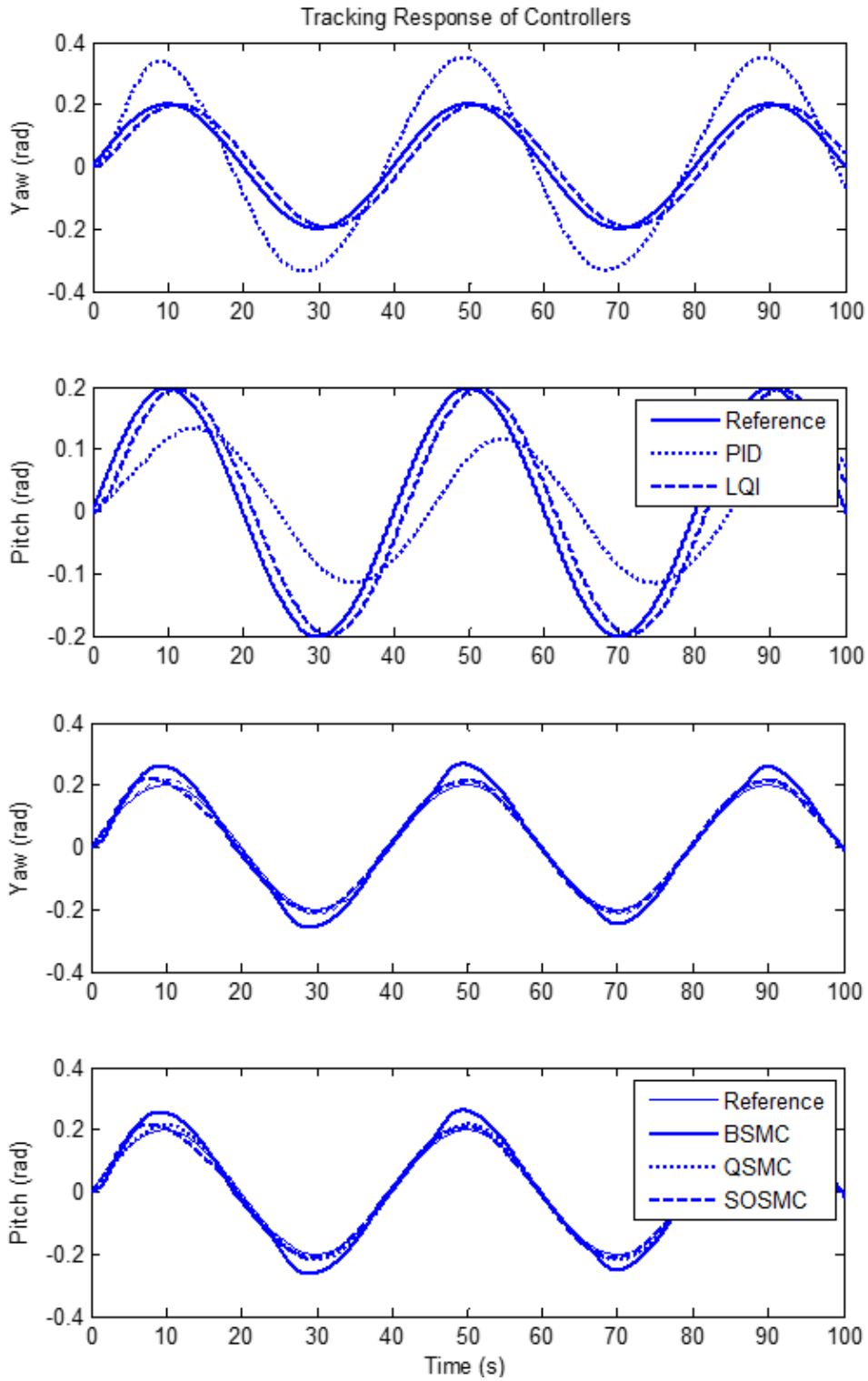

Figure 16: Comparison of simulated tracking response.

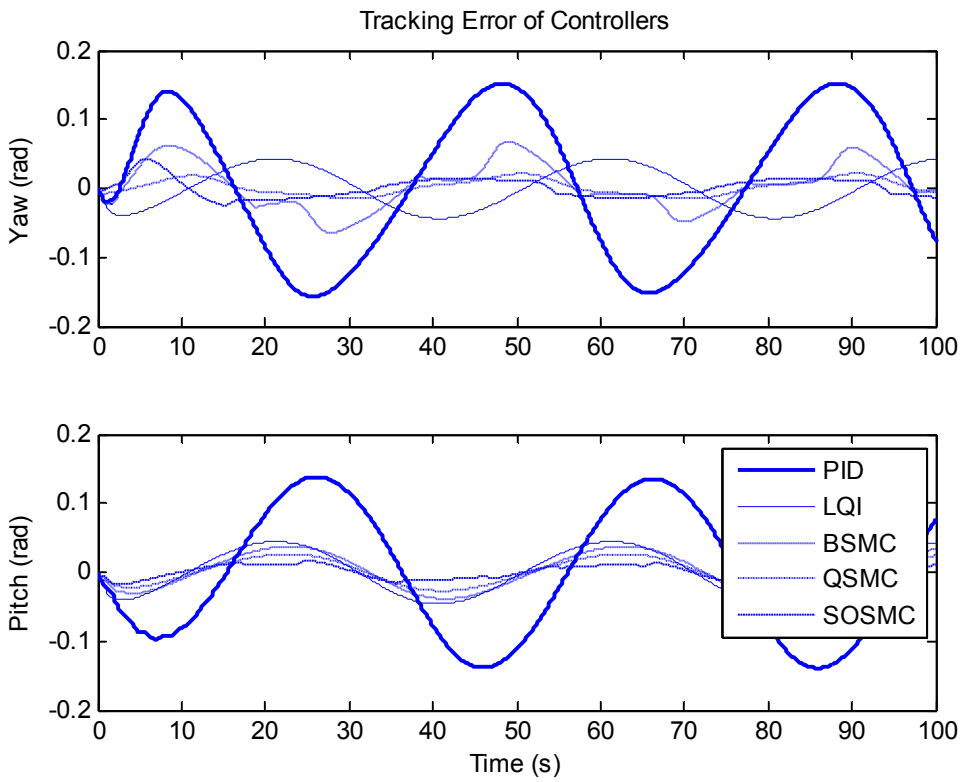

Figure 17: Comparison of simulated tracking error.

Table 2: Simulated MSE of the controllers.

Controller	Tracking Response	
	Pitch	Yaw
PID	28.9936	32.5295
LQI	9.8940	9.5308
BSMC	8.1661	10.5698
QSMC	5.7732	3.5881
SOSMC	2.9455	4.4467

Figure 18 shows the simulated control effort of the three SMC. The BSMC control effort shows some traces of chattering in the form of small spikes. However they are few and far in-between. Figure 19 shows a comparison of the switching gains.

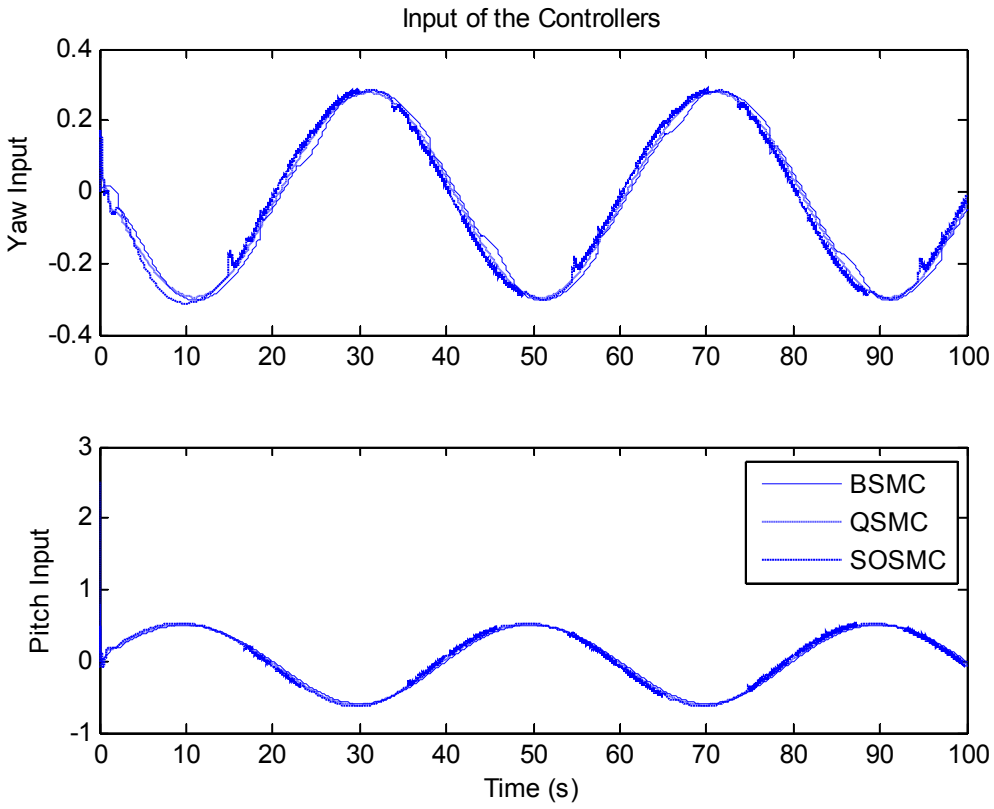

Figure 18: Comparison of simulated control effort.

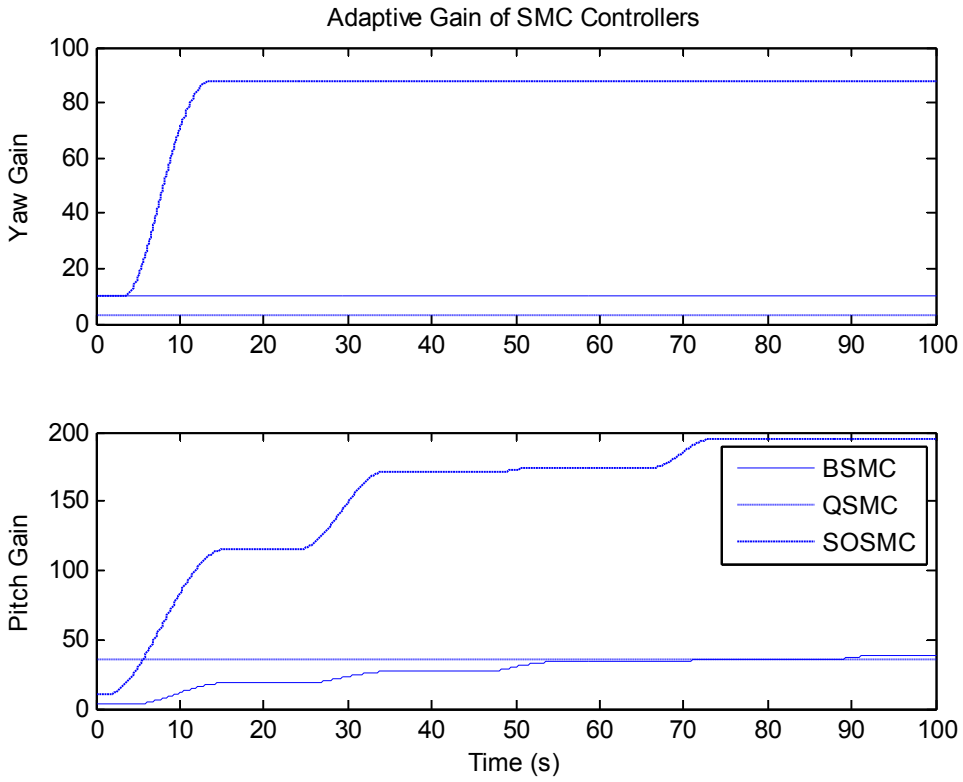

Figure 19: Comparison of simulated switching gain.

Figure 25 shows that the dead zone described before prevents the switching gain from growing to infinity. Another possible method is to clamp the adaptation law to at a specific amount, this is equivalent to placing an upper bound on it. A downside of this method is that if the limit has been reached on the switching gain and the system undergoes a disturbance or a change in parameter, that change cannot be accounted for.

5.A.iii. Comparison of Simulation Results

The simulation results of the LQI controller will be compared to the results obtained by Ankesh Kumar Agrawal [12]. Figure 20 shows the results obtained from A.K. Agrawal using bacterial forging to optimize the Q and R weighting matrices of a LQG controller in simulation. Figure 21 shows the results of the LQI controller in simulation for the same desired response. It is easy to see that the LQI controller provided superior response to the results obtained by A.K. Agrawal due to a decreased rise time and settling time. Additionally the simulation shows no overshoot and a rise time that is less than [12] by over 30 seconds.

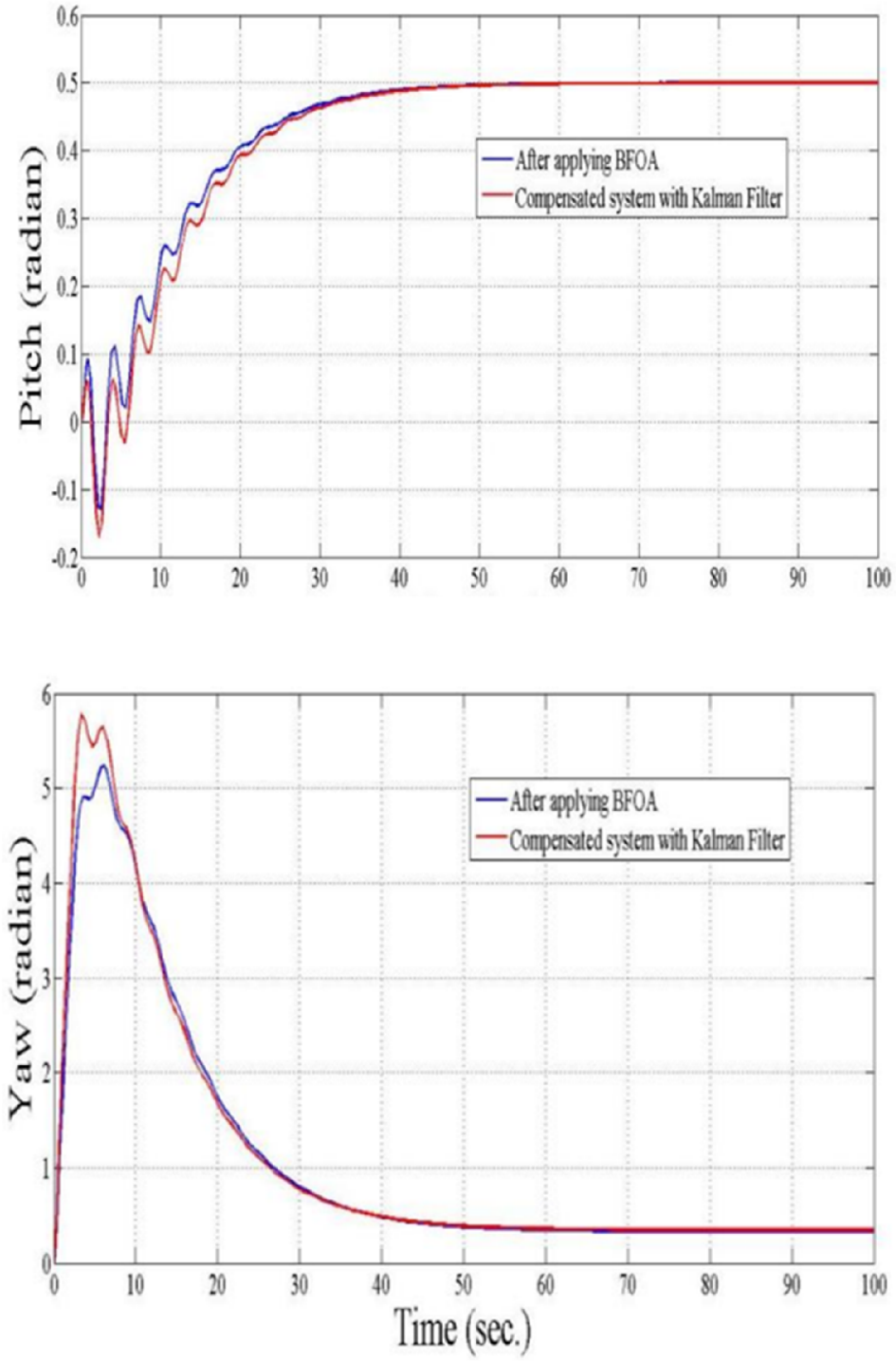

Figure 20: Simulated pitch and yaw step by A.K. Agrawal [12].

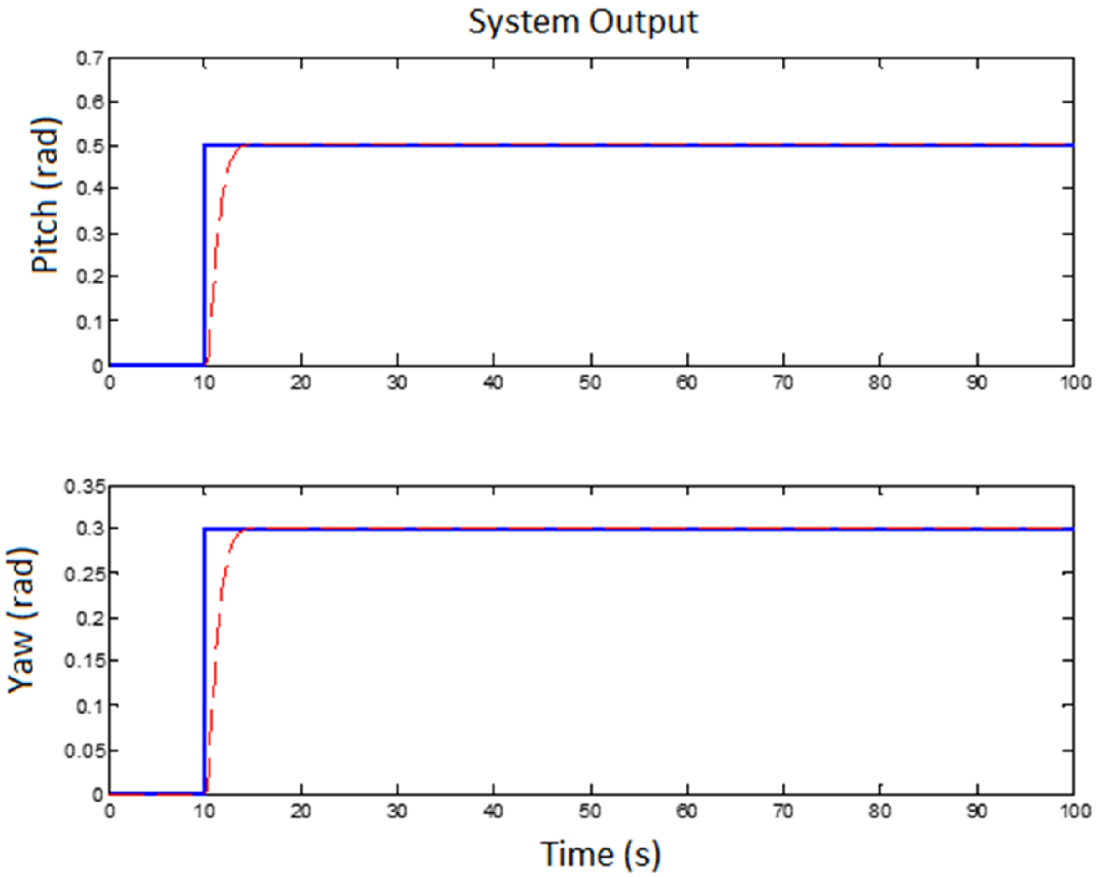

Figure 21: Simulated pitch and yaw step replicating results from [12].

The simulation results of the SMC controllers shown in this thesis are to be compared against those achieved by S. Mondal et al [10] next. The SOSMC designed here is based off of their work. Figure 22 shows the step response of the TRMS of the controller developed by [10]. Comparing this to Figure 13 it can be seen that the QSMC and SOSMC provide performance on par with that provided with the [10].

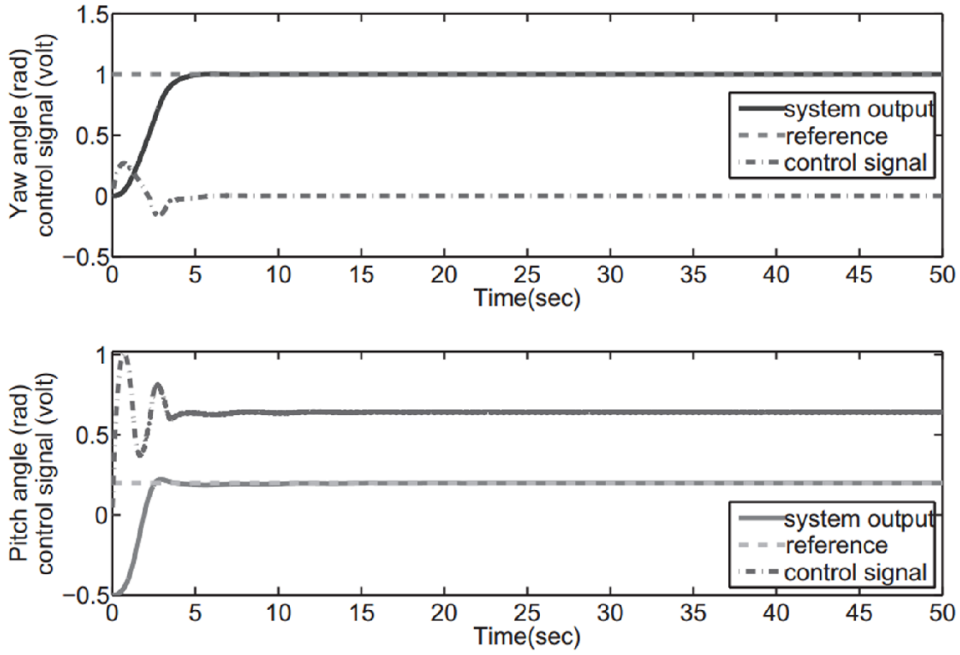

Figure 22: Simulated pitch and yaw step by S.Mondal et al [10].

Figure 23 shows the tracking response of their controller with a reference signal of $.02\sin(.025\pi t)$ for both inputs. Again comparing these results to those achieved in Figure 16 and Figure 17 it is shown that the QSMC and SOSMC provide performance on par with [10].

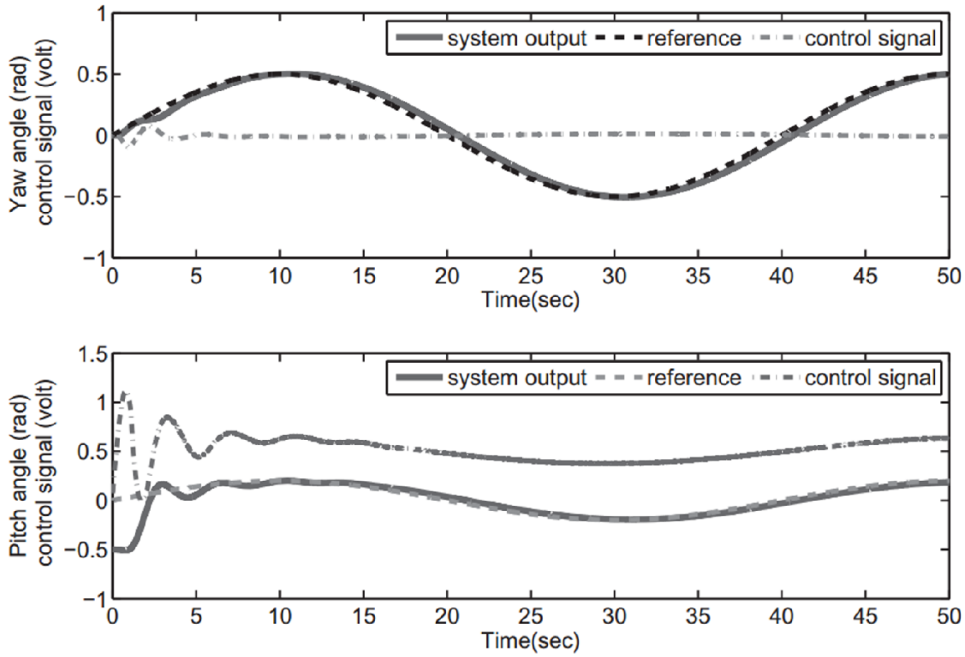

Figure 23: Simulated pitch and yaw tracking by S.Mondal et al [10].

5.B. Experimental Results

5.B.i. Step Response

The same reference signals were used in the experimental setup. The terms used for the LQI controller are given in Equations (128) and (129).

$$Q = diag([10^3; 200; 10; 10; 1; 1; 1; 10^3; 100]) \tag{128}$$

$$R = I \tag{129}$$

For the BSMC the parameters for the horizontal controller are defined as $C_H = [2, 1, 2]^T$, $D_H = [1\ 2\ 0]^T$, $\varphi_H = 1$, $\gamma_H = 4$, $\varepsilon_H = 3$, and the initial value of the switching gain is set to 10. The controller parameters for the vertical system are defined as $C_V = [5, 4, 6]^T$, $D_V = [20\ 10\ 0]^T$, $\varphi_V = 1$, $\gamma_V = 4$, $\varepsilon_V = 3$, and the initial value of the switching gain is set to 10.

The QSMC parameters for the horizontal controller are defined as $C_H = [3, 1, 2]^T$, $D_H = [.6\ 1.85\ 0]^T$, $\varphi_H = 1$, $\gamma_H = 4$, $\varepsilon_H = 3$, and the initial value of the switching gain is set to 10. The controller parameters for the vertical system are defined as $C_V = [5\ 4\ 6]^T$, $D_V = [15\ 11\ 0]^T$, $\varphi_V = .1$, $\gamma_V = 4$, $\varepsilon_V = 3$, and the initial value of the switching gain is set to 10.

The SOSMC parameters for the horizontal controller are defined as $C_H = [20\ 7\ .5]^T$, $D_H = [30\ 50\ 0]^T$, $\gamma_H = 6$, $\varepsilon_H = 4$, and the initial value of the switching gain is set to 10. The controller parameters for the vertical system are defined as $C_V = [5\ 4\ 2]^T$, $D_V = [30\ 15\ 0]^T$, $\gamma_V = 1$, $\varepsilon_V = 10$, and the initial value of the switching gain is set to 5.

Figure 24 shows the experimental step response of the controllers. As before the PID controller provided the worst performance, it had the most overshoot and the longest settling time. The LQI controller provided better performance, but still worse than the SMC. The QSMC and the SOSMC provided the best performance overall.

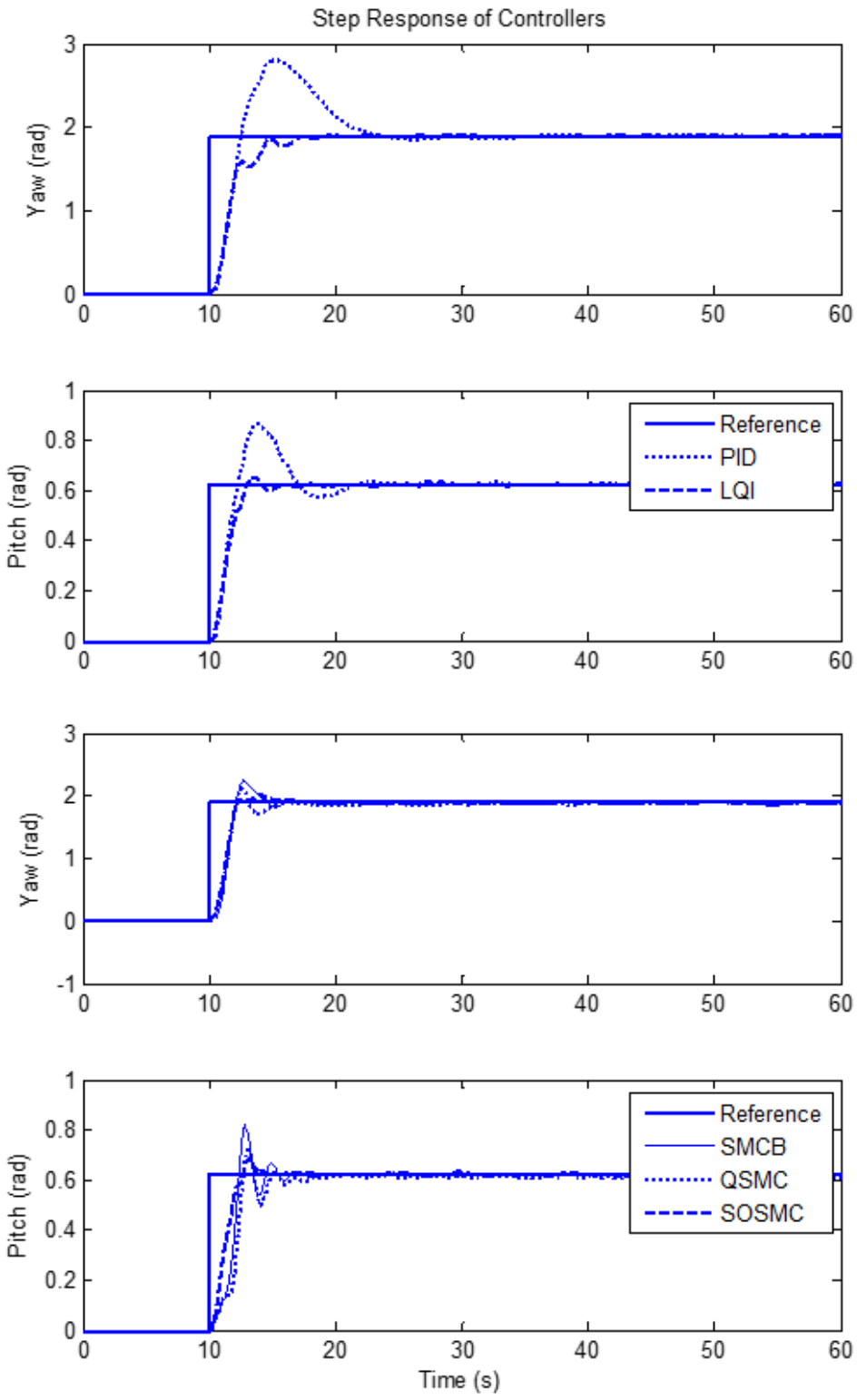

Figure 24: Comparison of experimental step responses.

Figure 25 shows the experimental switching gains of the SMCs. The BSMC and the QSMC converge to a similar switching gain, and the SOSMC converges to a larger value. There are two reasons this could happen. The first is that the scaling factor, $1/\gamma$, is larger. The second possible reason is that there is more uncertainty associated with the SOSMC due to the uncertainty block as shown in the derivation. The specific part of the uncertainty term that is responsible for the larger gain is the derivative of the ΔF term. It is expected that the derivative of the ΔF block will be larger because as Figure 26 shows there is still chattering present in the control effort that will make its way into the system by way of the observer.

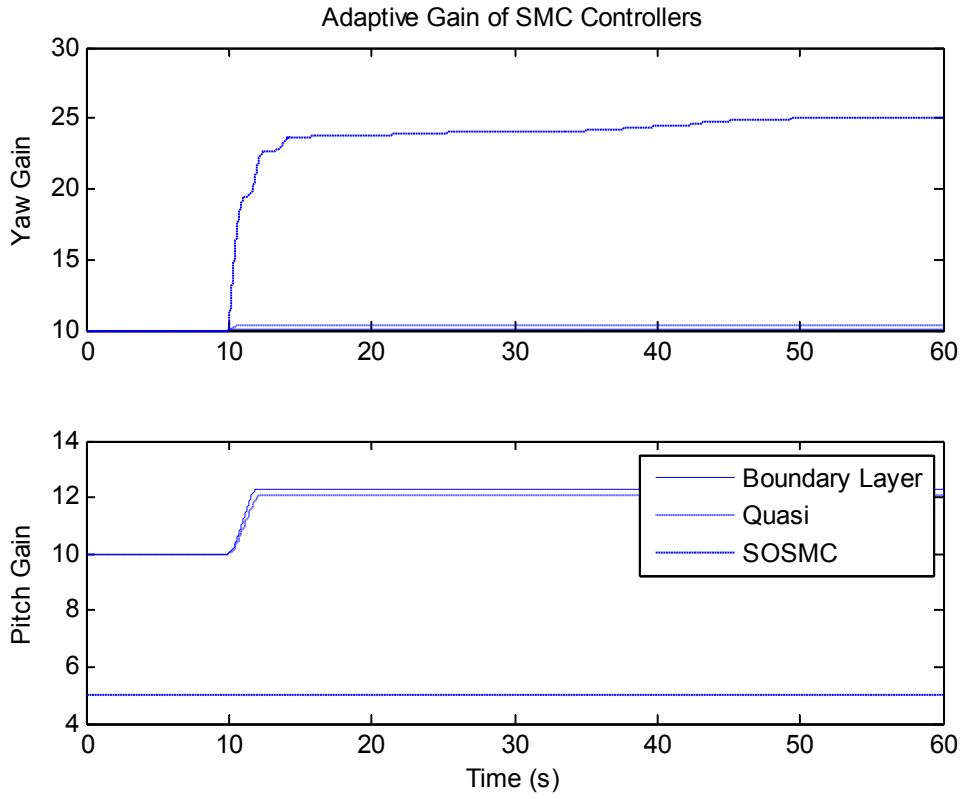

Figure 25: Comparison of experimental adaptive gain of the SMCs.

As Figure 24 shows all of the SMCs are able to provide good performance. This would lead one to expect that the chattering has been completely attenuated. However inspecting Figure 26 it can be seen that all three of the SMC have chattering present on the control signals.

The QSMC has the most chattering present, followed by the boundary layer and finally the SOSMC. The reason for the chattering is twofold. First the switching gain for the SOSMC is large, so even though the discontinuity is being integrated it can result in some chattering

making its way through. Even though the BSMC and the QSMC have smaller gains any chattering on the control signal will be effectively feedback into the system by way of the observer.

The second is because of the way the position feedback is provided. Position is measured using a rotary encoder with a resolution of .02 radians, (this was not replicated during simulation) position is measured by integrating the change over time. When the position changes it does so effectively instantly, in a step. This is measured from the output and fed back to the observer for state estimation. In order to replicate that output the estimated velocity must be large, and the result is a kind of chattering present on the estimated velocity term which is fed into the control law. Figure 27 shows a zoomed in version of the output from the observer using the SOSMC. The fast changes in the position due to the rotary encoder causes spikes in the estimated velocity. This is fed into the controller which causes chattering on the controller output.

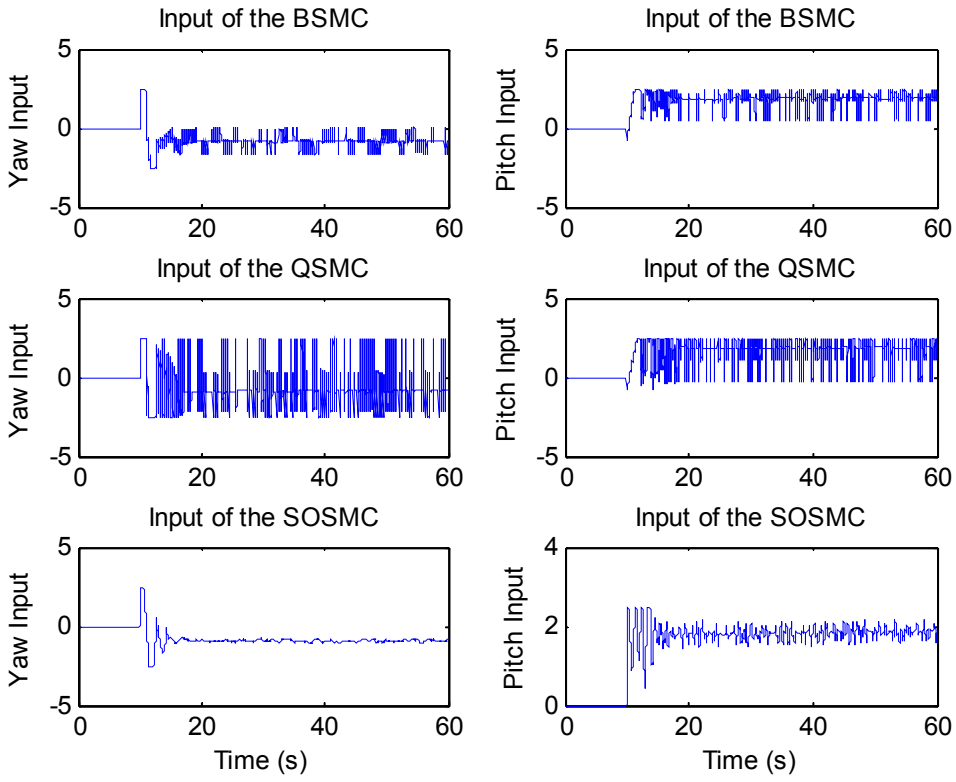

Figure 26: Comparison of experimental control effort of the SMCs.

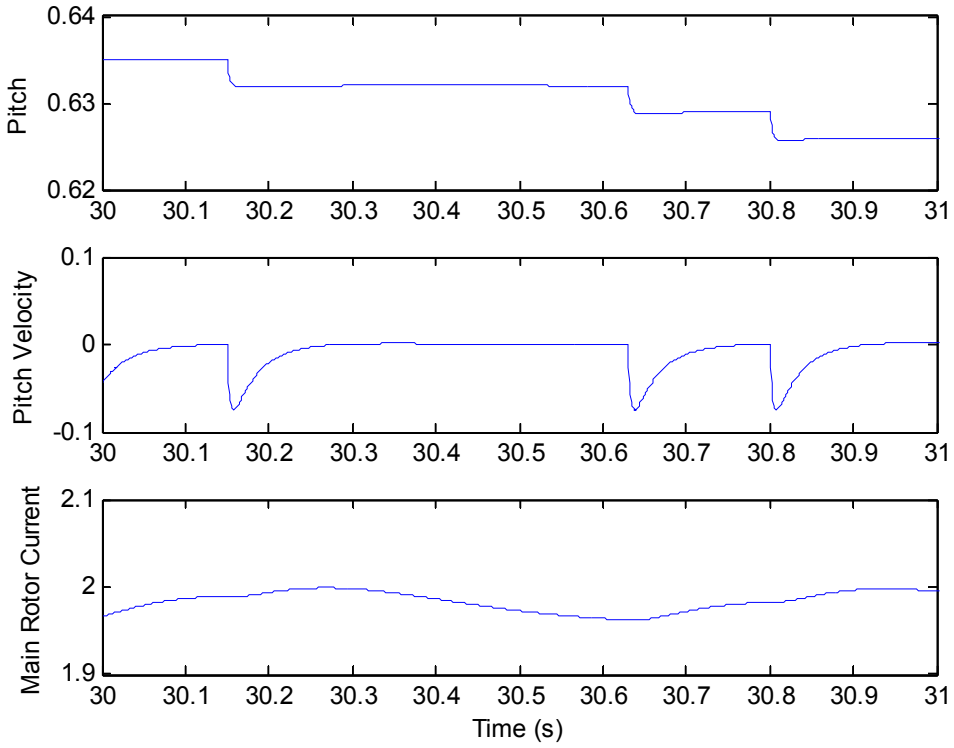

Figure 27: Output from the experimental observer with a step input with the SOSMC.

5.B.ii. Tracking Response

The same tracking signal was used in hardware as simulation. Figure 28 shows the tracking responses of the controllers. The PID controller was again the worst. It is easier to inspect performance by looking at the tracking error. Figure 29 shows the tracking error. The SOSMC provides the smallest error, the BSMC, QSMC and LQI all have about the same error. Table 3 shows the MSE of the experimental controllers.

Table 3: Experimental MSE of the controllers.

Controller	Tracking Response	
	Pitch	Yaw
PID	43.6562	28.4095
LQI	20.0760	13.3562
BSMC	21.3933	16.6058
QSMC	23.4072	18.6358
SOSMC	16.1903	15.8132

The PID controller had the worst performance. The SOSMC and the LQI controllers performed best, having the best tracking in the yaw and pitch respectively. An interesting trend between simulation and experiments is that the SOSMC consistently provides the best control for the pitch subsystem, but not for the yaw subsystem. It is possible that SOSMC isn't the best control method for the yaw subsystem, rather QSMC or LQI control is a better option. This could be due to the compromise that must be made to ensure good tracking performance as well as good a good step response.

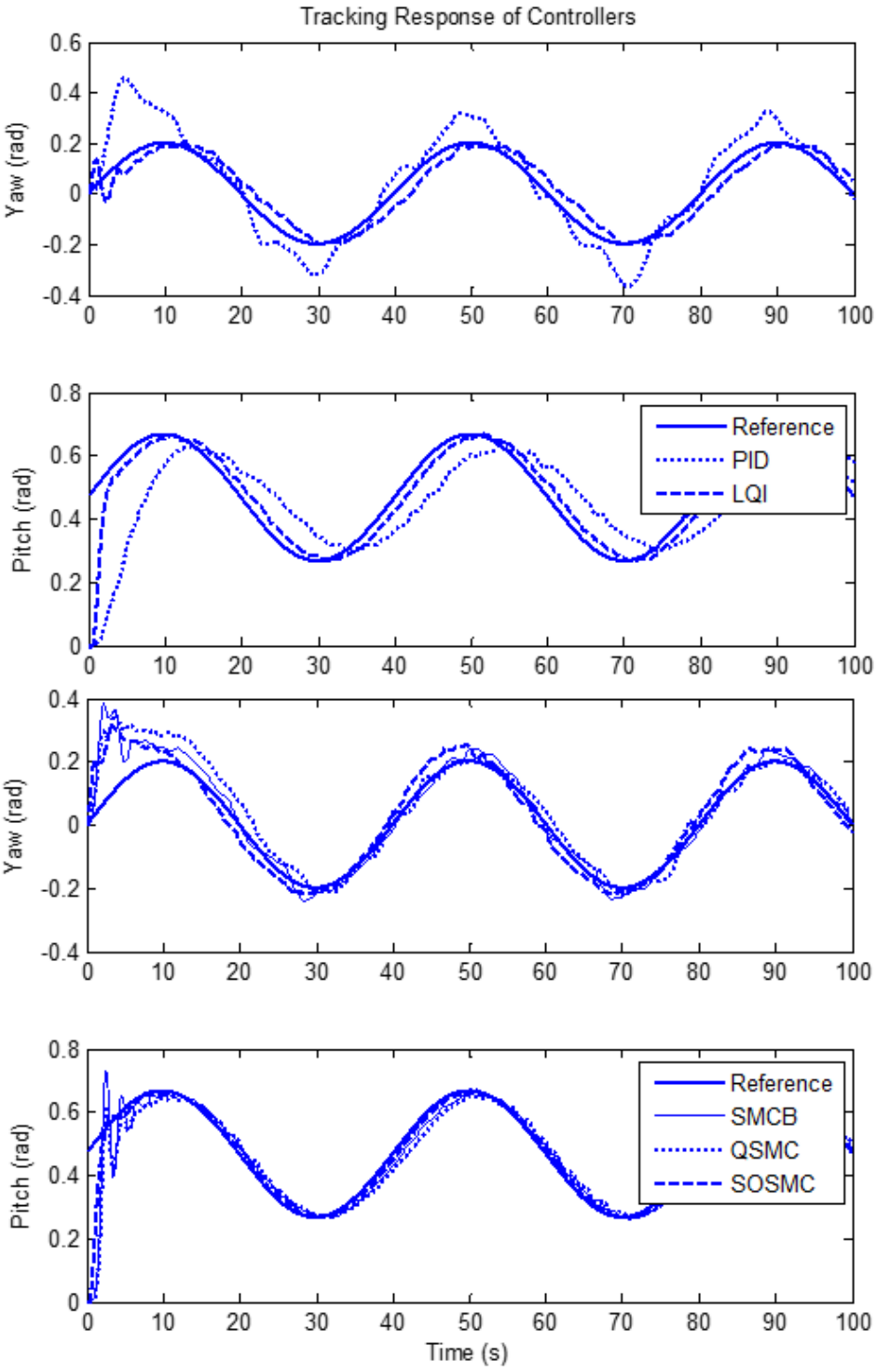

Figure 28: Comparison of experimental tracking responses.

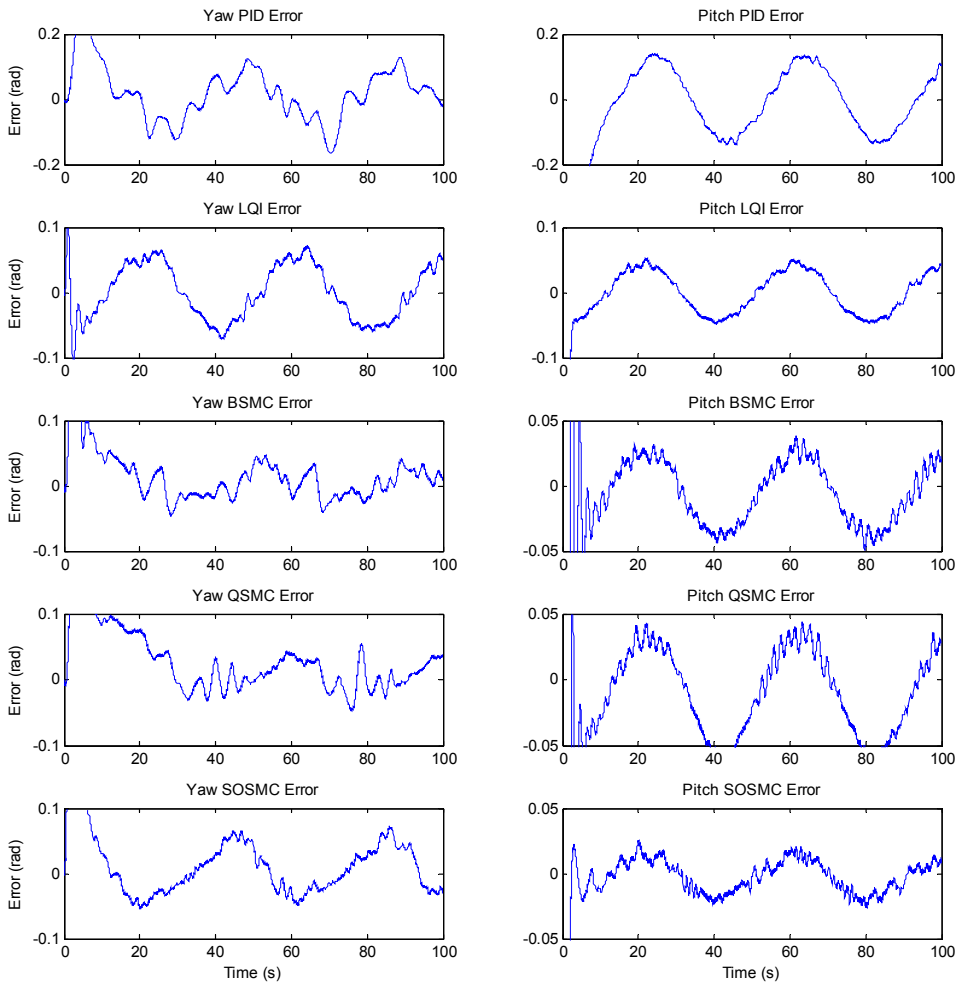

Figure 29: Comparison of experimental tracking errors.

Figure 30 shows the switching gain given a tracking signal. The switching terms converged to similar values as those from the step response. As before the SOSMC had a higher switching gain than the BSMC and QSMC.

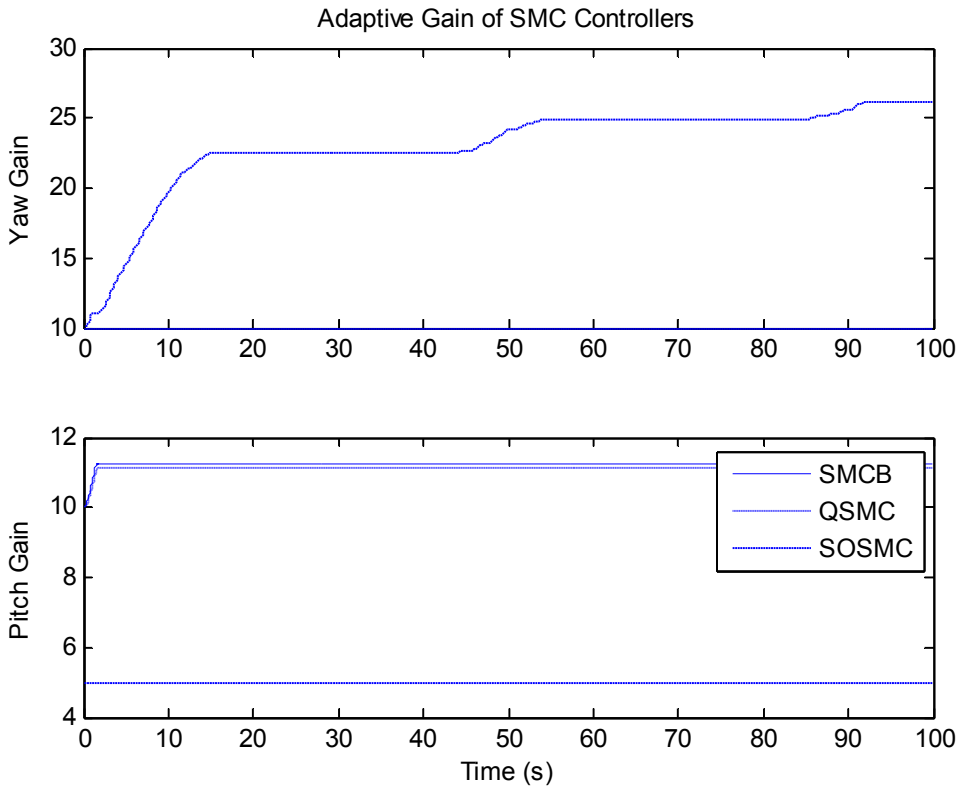

Figure 30: Comparison of experimental tracking adaptive gains.

Figure 31 shows the control effort of the three SMCs. Similar to the step response the QSMC had the most chatter, the BSMC had slightly less. Again as before the SOSMC had the least chatter of the three.

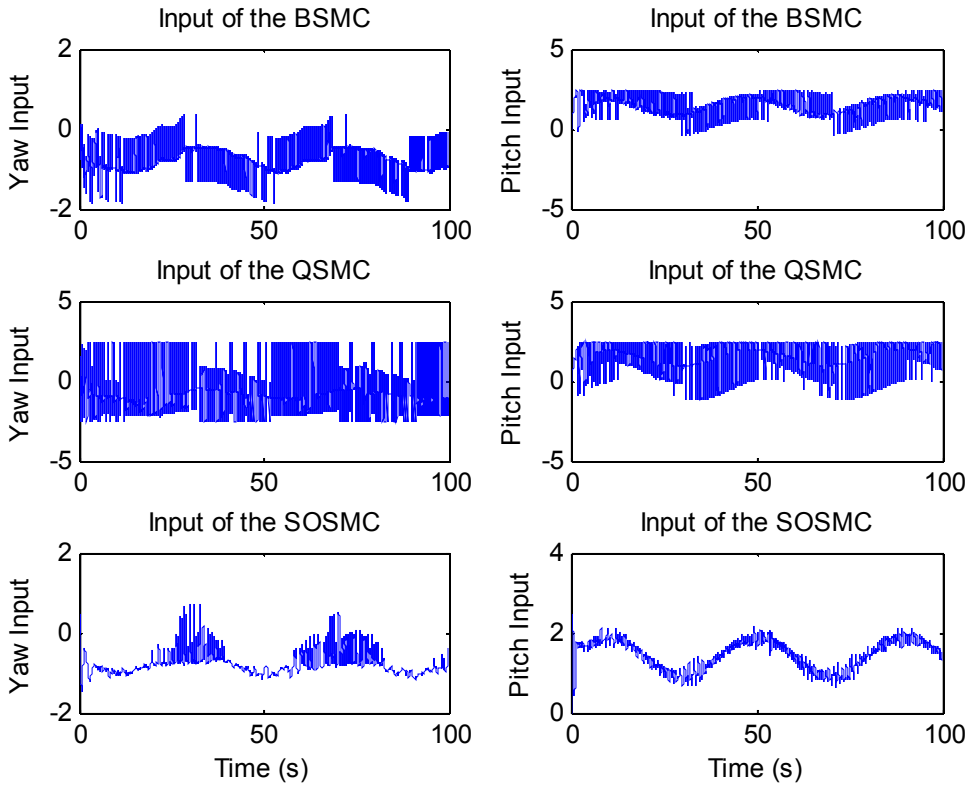

Figure 31: Comparison of experimental tracking control efforts.

5.B.iii. Comparison of Hardware Results

The hardware results shown here are going to be compared against those achieved by others that have implemented the controller experimentally. There is nothing in the literature about LQ based controllers applied to the TRMS being implemented in hardware. Because of this, the LQI controller will be compared against the same controllers as the SMC.

D.K. Saroj et al [14] implemented a SMC with a nonlinear state observer based off of the Luenberger structure. The experimental results were reported in the form of a step response. Figure 32 shows the step response of the controller.

It is seen that in Figure 32 the pitch subsystem has a settling time of about 20 seconds with no overshoot. The yaw subsystem has a settling time of about 15 seconds. From Figure 32, the SOSMC and the QSMC provide settling times of 15 seconds or less for the yaw subsystem. The LQI and SOSMC provide a settling time of less than 20 seconds for the pitch subsystem, and the QSMC and BSMC have settling times about equal to 20 seconds.

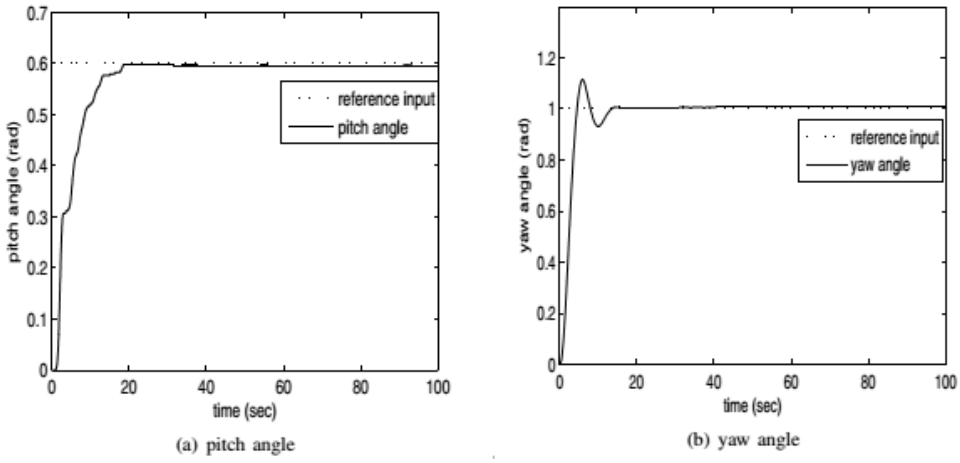

Figure 32: Experimental pitch and yaw step from D.K. Saroj et al [14].

There have been few tracking results published for the TRMS. In 2002 Juhng-Perng Su et al [33] published experimental results for the design and implementation of a terminal sliding mode controller (TSMC) for the TRMS system. Figures Figure 33 and Figure 34 show the tracking results of the TSMC.

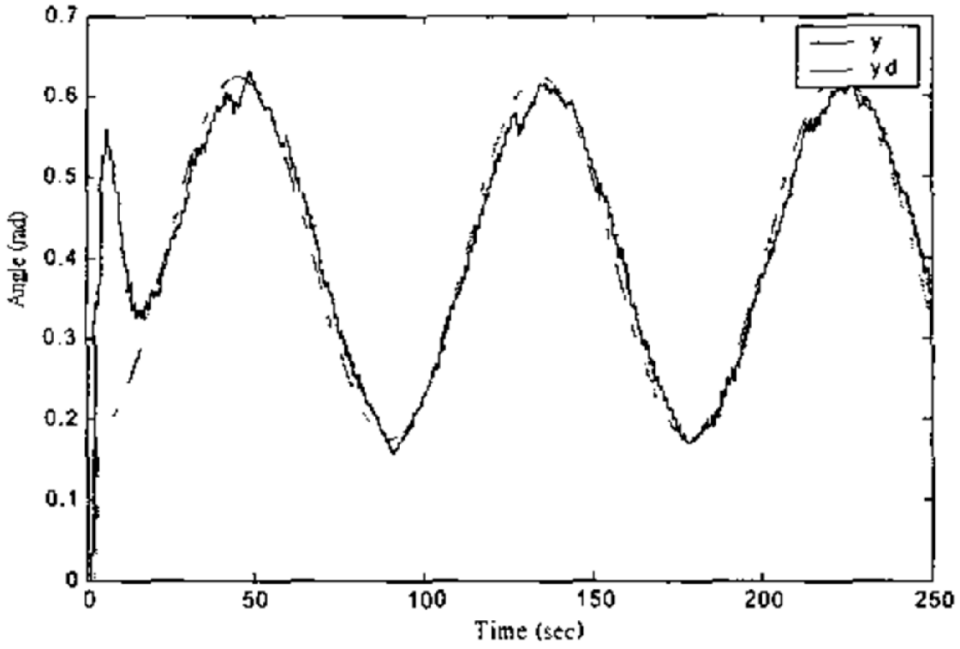

Figure 33: Experimental pitch tracking by Juhng-Perng Su et al [33].

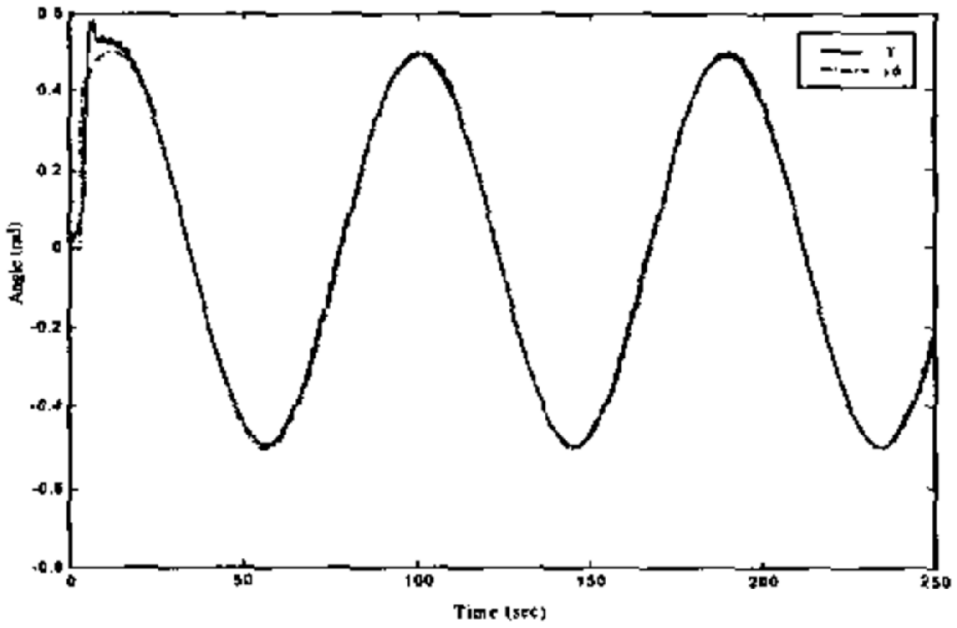

Figure 34: Experimental yaw tracking by Juhng-Perng Su et al [33].

It is simpler to investigate the performance by inspecting the tracking error. Figure 35 and Figure 36 show the tracking error for the TSMC. The tracking signal is different than the ones used before. The SMCs and LQI controllers were run again with a tracking signal of $0.5\sin(0.0225\pi t + 0.25\pi)$ for the yaw subsystem and $0.225\sin(0.0225\pi t - 0.5\pi)$ for the pitch subsystem. Figure 37 and Figure 38 show the results from the SMC and LQI controllers.

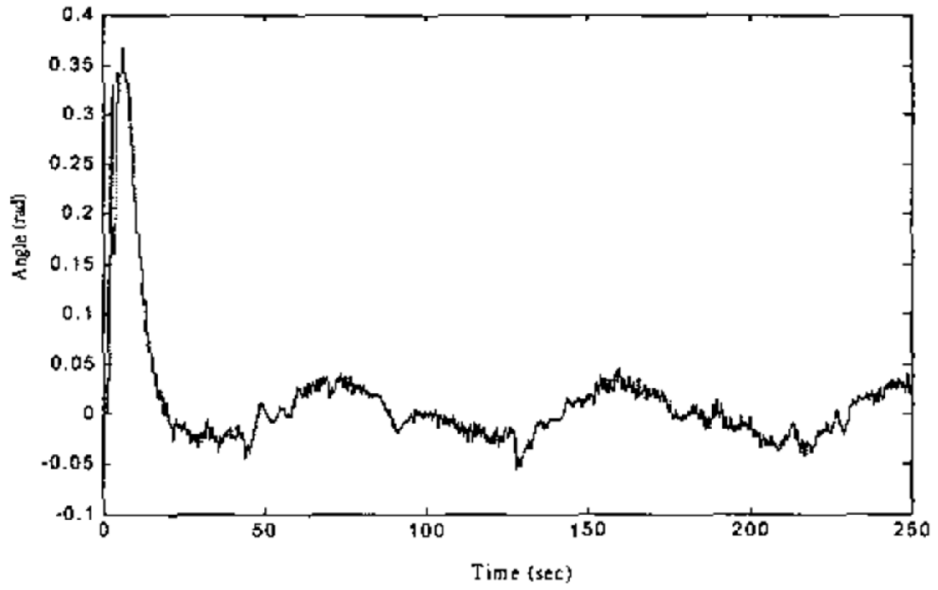

Figure 35: Experimental pitch tracking error by Juhng-Perng Su et al [33].

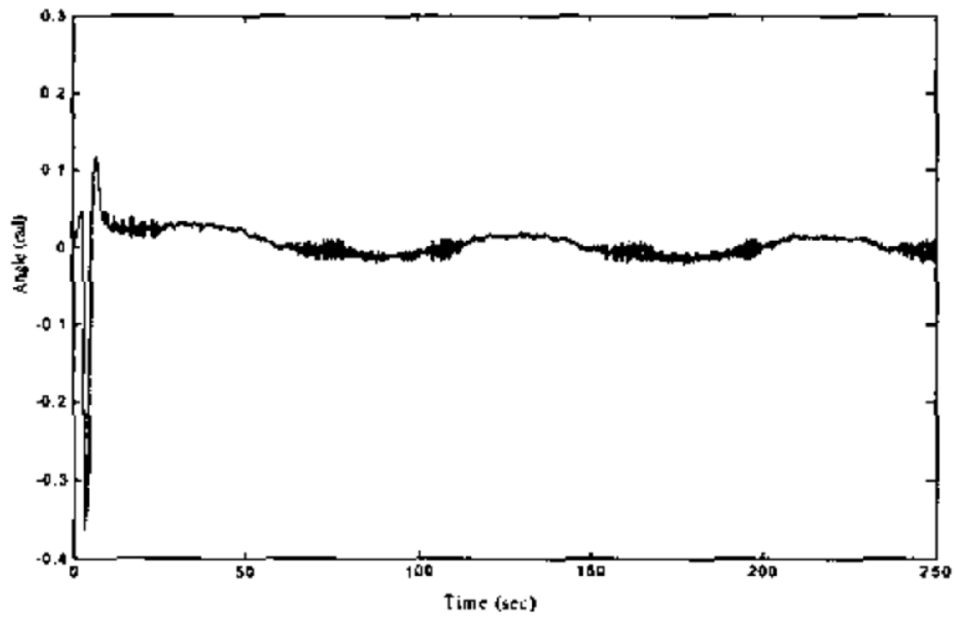

Figure 36: Experimental yaw tracking error by Juhng-Perng Su et al [33].

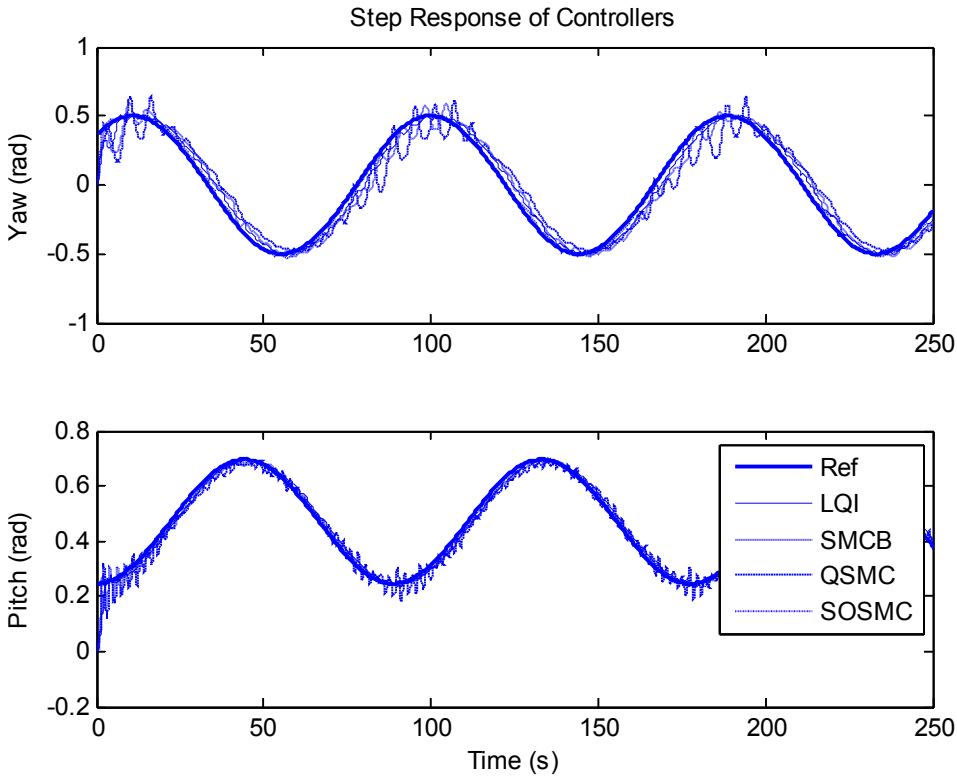

Figure 37: Experimental pitch and yaw tracking replicating results from [33].

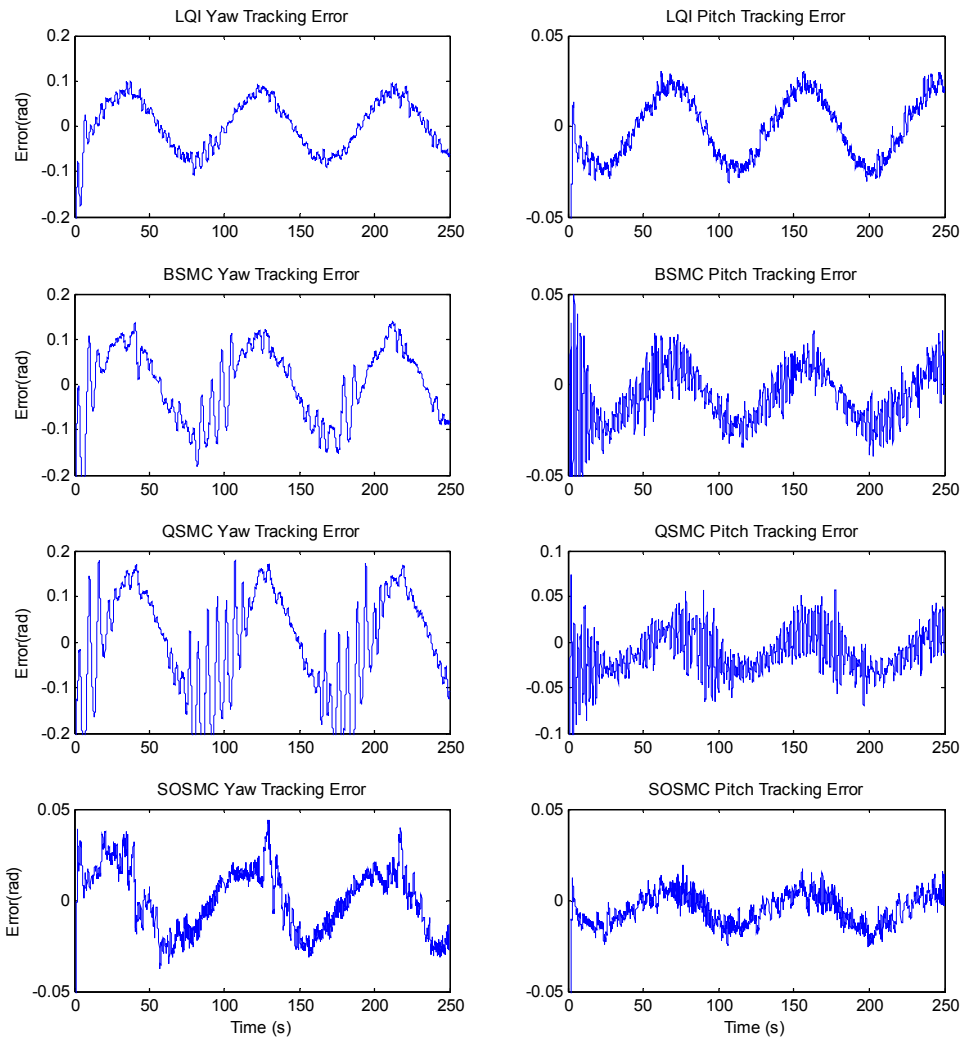

Figure 38: Experimental pitch and yaw error replicating results from [33].

The tracking error from the QSMC was worse than that achieved by Su et al for both pitch and yaw tracking due to small oscillations about the reference signal. These oscillations may be due to an increased sample period due to the limitations on MATLABs real time capabilities. MATLAB places a maximum on the number of data points than can be collected at 100,000. Because the simulation ran for 250 Hz, the minimum sample period is .0025 seconds, the controllers were designed for a sample period of .001 seconds. This wasn't an issue with the other controllers, but it probably was a problem for the QSMC.

The BSMC provided better tracking for the pitch subsystem but worse for the yaw for the same reason as the QSMC. The LQI and SOSMC provided superior tracking performance for both subsystems than the TSMC.

6 Conclusions

The LQI and SMC designed here showed superior performance to the existing controller solutions. In simulation it was shown that the LQI controller provided better performance than the LQG bacterial forging algorithm [12], and the QSMC and SOSMC provided performance that was just as good or performed as well as the adaptive higher order SMC that it is based on [10].

Experimentally it was found that the LQI and SMCs provided better step response than the existing SMC that have been applied experimentally. The tracking results showed the LQI and SOSMC provided the best performance, exceeding that of the TSMC of [33]. It was shown consistently that the SOSMC provided the best performance with minimal chattering. The downside is the more complicated control law.

There are several possible areas for future work. The first is an improved observer. A better observer, preferably a nonlinear one, would reduce the effect of the chattering that is fed back into the controller. This would reduce the effect of the chattering. Three possibilities are an unscented Kalman Filter, a sliding mode observer, or the nonlinear Luenberger observer used by [8].

The second area is a method that was explored earlier but was unable to have it work experimentally. The controller was an adaptive QSMC as before but instead of being designed using a linearized system, the controller was designed using feedback linearization, ignoring the cross coupling and treating them as disturbances.

Additional areas for future work involve optimization of the T_{xx} matrices of the SMCs and the switching gain. It may be possible to create an adaptation law to optimize the values of the T_{xx} matrices, rather than manually weighting them. As it was stated earlier, it is possible to calculate an optimal value of the switching gain knowing the bounds of the model parameters. The system may benefit from initializing the switching gain to this optimal value, and then letting the adaptation law increase the switching gain up to a bounded amount.

7 Works Cited

[1] M. Meon, T. Mohamed, M. Ramli, M. Mohamed and N. Manan, "Review and current study on new approach using PID Active Force Control (PIDAFC) of twin rotor multi input multi output system (TRMS)," in *2012 IEEE Symposium on Humanities, Science and Engineering Research*, Kuala Lumpur , 2012.

[2] S. P. Boyd, "EE363: Linear Dynamical Systems," 8 September 2012. [Online]. Available: http://www.stanford.edu/class/ee363/hw/hw7sol.pdf. [Accessed 13 January 2014].

[3] N. Lehtomaki, N. R. Sandell and M. Athans, "Robustness results in linear-quadratic Gaussian based multivariable control designs," *IEEE Transactions on Automatic Control,* vol. 26, no. 1, pp. 75 - 93 , 1981.

[4] C. Zhang and M. Fu, "A revisit to the gain and phase margins of linear quadratic regulators," *IEEE Transactions on Automatic Control,* vol. 41, no. 10, pp. 1527 - 1530 , 1996.

[5] A. Khan and N. Iqbal, "Modeling and Design of an Optimal Regulator for Three Degree of Freedom Helicopter/Twin Rotor Control System," in *Student Conference On Engineering, Sciences and Technology*, 2004.

[6] B. Pratap, A. Agrawal and S. Purwar, "Optimal control of twin rotor MIMO system using output feedback," in *2012 2nd International Conference on Power, Control and Embedded Systems (ICPCES)*, Allahabad , 2012.

[7] A. B. S. I. I. K. Q. Ahmed, "2-Sliding Mode Based Robust Control for 2-DOF Helicopter," in *International Workshop on Variable Structure Systems*, Mexico City, 2010.

[8] D. Saroj, I. Kar and V. Pandey, "Sliding mode controller design for Twin Rotor MIMO system with a nonlinear state observer," *International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing,* pp. 668 - 673 , 2013.

[9] S. Mondal and C. Mahanta, "Second order sliding mode controller for twin rotor MIMO

system," in *India Conference*, Hyderabad, 2011.

[10] S. M. C. Mahanta, "Adaptive second-order sliding mode controller for a twin rotor multi-input–multi-output system," *IET Control Theory and Applications,* vol. 6, no. 14, p. 2157–2167, 2012.

[11] S. Ahmad, A. Chipperfield and M. Tokhi, "Dynamic modeling and optimal control of a twin rotor MIMO system," in *Proceedings of the IEEE 2000 National Aerospace and Electronics Conference, 2000. NAECON 2000.*, Dayton, OH , 2000.

[12] A. K. Agrawal, "Optimal Controller Design for Twin Rotor," 2013.

[13] F. Allouani, D. Boukhetala and F. Boudjema, "Particle swarm optimization based fuzzy sliding mode controller for the Twin Rotor MIMO system," in *IEEE Mediterranean Electrotechnical Conference*, Yasmine Hammamet, 2012.

[14] D. Saroj, I. Kar and V. Pandey, "Sliding mode controller design for Twin Rotor MIMO system with a nonlinear state observer," in *International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing*, Kottayam , 2013.

[15] L. Huang, "An approach for robust control of a twin-rotor multiple input multiple output system," in *Robotics and Automation (ICRA)*, Shanghai, 2011.

[16] N. S. Nise, Control Systems Engineering, John Wiley and Sons, 2011.

[17] K. Ogata, Modern control engineering, Englewood Cliffs, N.J.: Prentice-Hall, 1970.

[18] F. Reis, R. Torrico-Bascope and M. Costa, "LQR control with integral action applied to a high gain step-up DC-DC converter," in *2011 Brazilian Power Electronics Conference (COBEP)*, Praiamar, 2011.

[19] F. L. Lewis, D. Vrabie and V. L. Syrmos, Optimal Control, John Wiley & Sons, 2012.

[20] J.-J. E. Slotine and W. Li, "Phase Plane Analysis," in *Applied Nonlinear Control*, New Jersey, Prentice Hall, 1991, pp. 17-39.

[21] J.-J. E. Slotine and W. Li, "Lyapunov Analysis of Non-Autonomous Systems," in *Applied Nonlinear Control*, New Jersey, Prentice Hall, 1991, pp. 105-116.

[22] J.-J. E. Slotine and W. Li, "Sliding Control," in *Applied Nonlinear Control*, New Jersey, Prentice Hall, 1991, pp. 276-310.

[23] "Intech," [Online]. Available: http://www.intechopen.com/source/html/17687/media/image63.png. [Accessed 08 02 2014].

[24] M. Dal and R. Teodoresci, "Sliding mode controller gain adaptation and chattering," *Turkish Journal of Electrical Engineering and Computer Sciences,* vol. 19, no. 4, pp. 531 - 549, 2011.

[25] C.-C. Fuh, "Variable Thickness Boundary Layers for Sliding Mode Control," *Journal of Marine Science and Technology,* vol. 16, no. 4, pp. 288 - 294, 2008.

[26] M.-S. Chen, Y.-R. Hwang and M. Tomizuka, "A State-Dependent Boundary Layer Design for Sliding Mode Control," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL,* vol. 47, no. 10, pp. 1677 - 1681, 2002.

[27] M. D. O'Toole, K. Bouazza-Marouf and D. Kerr, "Chatter Suppression in Sliding Mode Control: Strategies and Tuning Methods," in *Robust Design, Dynamics and Control*, Springer Vienna, 2010, pp. 109-116.

[28] W. L. Brogan, "Observers-Reconstructing the State from Available Outputs," in *Modern Control Theory*, New Jersey, Prentice Hall, 1991, pp. 431-473.

[29] Feedback Systems Ltd, "Twin Rotor MIMO Teaching Manual 1," [Online]. Available: http://www.feedbacksystems.com/. [Accessed 10 September 2013].

[30] F. I. Inc, "http://www.feedback-instruments.com/products/education/control_instrumentation/twin_rotor_mimo," [Online]. Available: http://www.feedback-instruments.com/products/education/control_instrumentation/twin_rotor_mimo. [Accessed

10 September 2013].

[31] F. I. Inc, "Twin Rotor MIMO," [Online]. Available: http://www.feedback-instruments.com/products/education/control_instrumentation/twin_rotor_mimo. [Accessed 10 September 2013].

[32] G. J. Bierman, Factorization Methods for Discrete Sequential Estimation, Mineola: Dover Publications, 2006.

[33] J.-P. Su, C.-Y. Liang and H.-M. Chen, "Robust control of a class of nonlinear systems and its application to a twin rotor MIMO system," *2002 IEEE International Conference on Industrial Technology,* vol. 2, pp. 1272-1277, 2002.

[34] R. Schkoda, "Dynamic inversion of underactuated systems via squaring transformation matrix," 2007.

Appendix A: Kalman Gain over Time

```
%% Discrete time system and initial conditions
A = [.5183 .1173;-.5867 .5183];
B = [5*exp(.2)-1;-exp(-.1)-1];
x1(:,1) = [0;5];
k = 0:1:20;

%% Open Loop simulation
for i = 1:20
    x1(:,i+1) = A*x1(:,i);
end

figure(1)
plot(k,x1(1,:),k,x1(2,:),'r');
hold on
title('Comparison of States');
xlabel('Time');
ylabel('State Values');

%% Full LQR Control
Sn = 5*eye(2);
Q = [2 1;1 2];
R = 1;
S(:,:,21) = Sn;
x2(:,1) = x0;
K1 = zeros(20,2);

for i = 20:-1:1
    S(:,:,i) = A.'*S(:,:,i+1)*(eye(2)-B/(B.'*S(:,:,i+1)*B+R)*...
        B.'*S(:,:,i+1))+Q;
end

for i = 1:20
    K1(i,:) = (R-B.'*S(:,:,i+1)*B)\B.'*S(:,:,i+1)*A;
    x2(:,i+1) = (A+B*K1(i,:))*x2(:,i);
    U(i,:) = -K1(i,:)*x2(:,i);
end

plot(k,x2(1,:),'c',k,x2(2,:),'m');
```